

ASP 3

ובניית אתרים ב - Visual InterDev

סדנת לימוד

מהדורה שנייה



כולל Visual InterDev
גירסה מלאה ל-90 יום

רונן אלמוג

ASP 3
ובניית אתרים ב-
Visual InterDev

סדנת לימוד

מהדורה שנייה

עקוב במהלך הספר אחר הוראות ההתקנה וההפעלה של התקליטור

עורך ראשי ומקצועי: **זהר עמיהוד**

עריכה לשונית ועיצוב: **טליה טופז, שרה עמיהוד**

עיצוב עטיפה: **סטודיו מצגר**

שמות מסחריים

שמות המוצרים והשירותים המוזכרים בספר הינם שמות מסחריים רשומים של החברות שלהם. הוצאת הוד-עמי עשתה כמיטב יכולתה למסור מידע אודות השמות המסחריים המוזכרים בספר זה ולציין את שמות החברות, המוצרים והשירותים. שמות מסחריים רשומים (registered trademarks) המוזכרים בספר צוינו בהתאמה.

הודעה

ספר זה מיועד לתת מידע אודות מוצרים שונים. נעשו מאמצים רבים לגרום לכך שהספר יהיה שלם ואמין ככל שניתן, אך אין משתמעת מכך כל אחריות שהיא.

המידע ניתן "כמות שהוא" ("as is"). הוצאת הוד-עמי אינה אחראית כלפי יחיד או ארגון עבור כל אובדן או נזק אשר ייגרם, אם ייגרם, מהמידע שבספר זה, או מהתקליטור שמצורף לו.

לשם שטף הקריאה כתוב ספר זה בלשון זכר בלבד. ספר זה מיועד לגברים ונשים כאחד ואין בכוונתנו להפלות או לפגוע בציבור המשתמשים/ות.

- **טלפון: 09-9564716**
- **פקס: 09-9571582**
- **דואר אלקטרוני: info@hod-ami.co.il**
- **אתר באינטרנט: http://www.hod-ami.co.il**

ASP 3
ובניית אתרים ב-
Visual InterDev

סדנת לימוד

מהדורה שנייה

רונן אלמוג



ASP 3 and Site Development with Visual InterDev, 2Ed.

By

Ronen Almog

Editor: **Z. Amihud**

הוצאת הוד-עמי

לספרי מחשבים בע"מ

ת.ד. 6108 הרצליה 46160

טלפון: 09-9564716 פקס: 09-9571582

www.hod-ami.co.il

הודפס בישראל 2001

HOD-AMI Ltd.

P.O.B. 6108, Herzliya

ISRAEL, 2001

ISBN 965-361-267-0 מסת"ב

תוכן עניינים מקוצר

נספחים (מתחיל בסוף הספר)

תוכן העניינים

נספחים

הקדמה

איך ספר ASP נולד? כמו תינוק. טוב, כמעט. בשנתיים האחרונות לימדתי הרבה מאוד אנשים בקורסים שונים של E-Commerce, VID, Web Masters, ASP, ובניית אתרים. היעדרו של ספר טוב בעברית מאוד הפריע לי - ספר כמו שאני אוהב לקרוא - פשוט, קריא, מלא דוגמאות, לא תיאורטי מדי, קצת משעשע לפעמים, ושיעזור לי לבנות את האתר שלי. אחרי שגמרתי לקטר, התחלתי לכתוב. את כל מערכי השיעור שלי הכנסתי לספר הזה, ועוד קצת. **זהו קורס שלם לפיתוח אתרי ASP בעזרת Visual InterDev**. לפתח אתרי ASP זה כיף. בכלל, לבנות משהו פועל וחי שהוא שלך – זה כיף. אמרתי, כמו תינוק. לפחות האתר שלכם לא יבכה בלילה...

הקמתי **אתר במיוחד לספר זה**, ואתם מוזמנים לבקר בו: **www.sifroni.co.il**. באתר יש כמה פעילויות:

א. יש בו דוגמה מלאה ופועלת לחנות הספרים "ספרוני" שבנייתה מודגמת בספר זה. כדאי לראות איך זה צריך להיראות ולתפקד. אתם יכולים להיכנס ו"לקנות" ספרים ללא חשש. זו רק דוגמה, לא תחויבו בתשלום ושום דבר לא יישלח אליכם בדואר.

ב. פורום תמיכה. בפורום זה אני מנסה לעזור לכם בשאלות שנוגעות לספר, ולפעמים גם טיפה מעבר לו. אתם מוזמנים גם לעזור זה לזה. מובן שאין זו הזמנה ללימוד ASP בהתכתבות. פנו אלי רק במקרים קשים שלא הצלחתם לפתור לבד, ולאחר שעיינתם בפרק "איתור תקלות". לפני שליחת שאלה, אנא חפשו מעט בפורום. רוב השאלות כבר נשאלו וקיבלו תשובה. אין טעם לענות שוב ושוב על אותה שאלה. עיינו גם בדף "שאלות נפוצות", בו מרוכזות השאלות הפופולריות ביותר.

ג. וכמובן: קישורים מעניינים לאתרים שעוסקים ב-ASP ובניית אתרים, טיפים ועוד.

אין לי ספק שתפיקו תועלת רבה מן הספר הזה, אבל אני מקווה ליותר. אני מקווה שתפיקו ממנו גם הרבה הרבה הנאה.

תודה לסיגל על הסבלנות, תודה לשירי (מערכות מידע) ולאורי, תודה לגילי שזחלה בסביבה ועשתה Reset למחשב שלי מדי פעם, ותודה לכל התלמידים ששולחים לי שאלות בפורום ומראים לי את כל האתרים המדהימים שבנו. זה הכל בשבילכם.

בהצלחה לכולם!

רונן אלמוג

מה בתקליטור

במהלך הלימוד בספר תמצאו הוראות התקנה והתייחסות לתקליטור המצורף.

בנוסף ל- Visual InterDev ו- Personal Web Server, מכיל התקליטור גם את כל פרויקט חנות הספרים "ספרוני" הנבנית במהלך הספר.

מומלץ לעבוד לפי ההוראות בספר, ולא להעתיק את קבצי ASP מהתקליטור. כתיבה אמיתית תיתן לכם תחושה אמיתית וניסיון יותר מאשר העתקת קבצים. אם לא עובד לכם קובץ מסוים, אולי יהיה לכם נוח להשוות לקובץ בדוק ועובד. את קבצי התמונות והלחצנים העתיקו בכל מקרה מהתקליטור, כי אין טעם ליצור קבצים חדשים. כך גם תהיה זהות של השמות בין קבצי התמונה לבין המוזכר בתוכניות.

מצאו בתקליטור את התיקיה Bookstore. בתוכה נמצאים כל הקבצים המרכיבים את הפרויקט.

בתיקיה Bookstore\ASP_Code נמצאים קבצי המקור. לאחדים מן הקבצים יש יותר מגירסה אחת, לפי התפתחות הכתיבה בספר. בכל שלב בדיון מוזכר שם הקובץ בו תמצאו את קוד המקור נכון לאותו שלב.

בתיקיה Bookstore\Images נמצאים כל הלחצנים, הבאנרים ושאר האלמנטים הגרפיים השייכים לפרויקט. במהלך בניית הפרויקט תעבירו מתיקיה זו קבצים לפי הצורך לתיקיה Images בפרויקט.

בתיקיה Bookstore\Buttons נמצאים הלחצנים השייכים לתפריט. במהלך בניית התפריט תיצרו תיקיה חדשה עבורם בשם זה - Buttons - בתוך הפרויקט, ותעתיקו אליה את הקבצים.

בתיקיה Bookstore\DataBase נמצא קובץ יחיד: bookstore.mdb, מסד הנתונים של הפרויקט. גם כאן, מומלץ לבנות אותו בעצמכם לפי ההוראות שבספר. במקרה שהסתבכתם – קחו את הקובץ מהתקליטור. מיקומו במחשב שלכם אינו חשוב, ניתן לשמור אותו בתיקיית הפרויקט.

בתיקיה Software תמצאו מספר תוכנות עזר:

ACDC – תוכנה לצפייה וניהול של קבצים גרפיים. חובה.

ASPEdit – עורך פשוט ונחמד, בעל יכולות ASP.

HomeSite 4.5 – אחד מכלי בניית האתרים הוותיקים והמוצלחים ביותר. תומך כמובן גם ב-ASP.

ScriptWorx – כלי יפה וחזק לבניית אתרים, חינם לגמרי. תומך בשפות רבות: ASP, PHP, DHTML ועוד. מכיל תבניות מוכנות מראש, נוח מאוד לעיתים.

WinSock – שדרוג ל- Winsock 2.0. דרוש להתקנת PWS ב- Windows 95.

פרק 1

מה זה ASP

משהו השתנה פה!

מעניין אם רבים מגולשי האינטרנט שמו לב לתופעה, אבל שינוי עצום מתחולל בעולם האינטרנט בשנה האחרונה: עד ממש לא מזמן, מי שרצה לכתוב אתר אינטרנט כתב אותו ב-HTML. בדרך כלל הוא אפילו השתמש באחד המחוללים שקמו לאלפיהם, ואפילו HTML הוא לא היה ממש חייב לדעת. כמה תגים פה, GIF Animation שם, רקע מגניב, ו..הופ, יש אתר. היו ימים..., כמה התגאינו באתרים שבנינו.

לפתע התחיל מבול של אתרים מסוג אחר לגמרי. אתרים פעילים, אינטראקטיביים, חכמים, וגם שימושיים. אתרים שלא מסתפקים בלהיות סתם לוח מודעות, אלא מנהלים דו-שיח עם הגולש, מספקים לו מידע לפי דרישתו, מבצעים את בקשותיו, מוכרים לו מוצרים, בקיצור: תוכניות מחשב לכל דבר.

בתחילת דבריי היה צליל של זלזול ב-HTML, והקורא העירני יכול לשאול: הייתי באתר amazon.com, אתר ממוחשב ומפותח לכל הדעות, בחרתי בתפריט View > Source, וראיתי כיצד בנוי הדף. זה היה HTML, לא ASP.

שאלה חשובה, וטוב ששאלתם אותה. אנחנו אמנם רוצים לבנות תוכנית מחשב שתמכור ספרים, אבל מה לעשות שקהל הצרכנים שלנו חמוש בדפדפן בלבד, ודפדפן, כידוע לכל, יודע להבין רק HTML. לכן, ניצור תוכנית שמייצרת HTML. לדוגמה, בקשו מאתר amazon את כל הספרים שמכילים את המילה "שוקולד". תקבלו רשימה כלשהי. בחרו View > Source ותראו דף HTML מוכר וחביב, עם טבלה ובה שורה עבור כל ספר. מה זה אומר? שיש להם דף כזה מוכן עם כל הספרים בהם מוזכר שוקולד? התשובה היא: לא. התוכנית בנתה אותו **דינמית**, במיוחד בשבילכם! התוכנית סרקה את כל הספרים במאגר, דירגה אותם לפי ההתאמה לדרישותיך, בנתה בזמן אמת קובץ HTML ושלחה אותו אליכם! לעולם לא תראו את תוכנית ה-ASP שבנתה את הדף. אתם רואים רק את **תוצאותיה.**

אז מה זה דף ASP?

דף ASP הוא קובץ טקסט רגיל, בעל סיומת ASP, המכיל שילוב של HTML ופקודות תכנות.

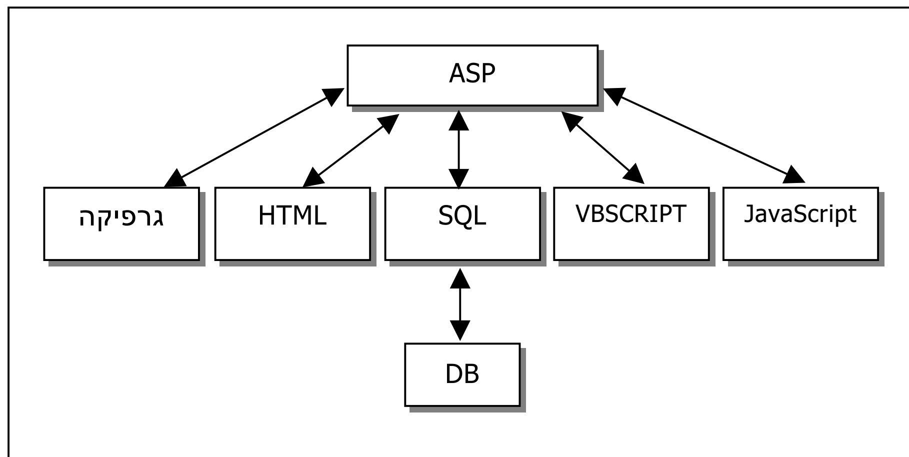
כאשר מבקשים משרת אינטרנט דף HTML, הוא פשוט שולח אותו לדרכו. כאשר מבקשים ממנו דף ASP, הוא מתנהג אחרת לגמרי. השרת נכנס לקובץ ומתחיל לפענח אותו. תגי HTML רגילים נשלחים לגולש (לדפדפן). פקודות תוכנה מתחילות להתבצע. בהמשך הספר נראה דוגמאות רבות לכך.

למי מיועד ספר זה?

ספר זה נכתב במיוחד עבור בוני אתרים. אם כבר בניתם אתר סטטי ואתם רעבים ליותר, זה הספר בשבילכם. בספר זה תלמדו לשלב מסדי נתונים באתר. תמצאו בו אוצר של ידע וטכניקות עבודה, שיקפיצו אֶתכם מדרגה לעבר אתר דינמי, תוסס ומלא פעילות. אתר שלא רק מציג משהו לגולש, אלא מקיים איתו דיאלוג. אתרי מכירות, מכרזים, חדשות, משחקים, הכל נבנה היום בעזרת טכנולוגיית ASP. אתר הדוגמה שנבנה לאורך הספר הוא "ספרוני" - חנות ספרים וירטואלית. תוך כדי הבנייה ולימוד השפה, נדונים גם שיקולים רבים, הקשורים להקמת עסק באינטרנט: בחירת מסד נתונים, בניית מסד נתונים, עבודה מול מסד נתונים, יצירה וניהול של סל קניות, קבלת כרטיסי אשראי, אבטחה, טכנולוגיית SSL ועוד.

כתיבת אפליקציית אינטרנט ב-ASP היא משימה רב תחומית. עליכם לדעת כמובן ASP, אבל זה לא מספיק. ככותבי תוכניות המייצרות קבצי HTML, עליכם כמובן לשלוט היטב ב-HTML בעצמכם. לא יודעים HTML כלל? אל תקנו ספר זה! (אני ממליץ על הספר "**HTML 4 למפתחי אתרים באינטרנט**", בהוצאת **הוד-עמי**). גם מעט VBscript יכול לעזור. שפת SQL היא השפה בה עובדים מול מסדי נתונים, ואנו נשתמש בה רבות. לא ממש אלמד SQL בספר זה, ישנם ספרים מצוינים לכך (אני ממליץ על הספר "**בסיסי נתונים טבלאיים ושפת SQL**" בהוצאת **הוד-עמי**). עליכם להיות מסוגלים לבנות מסד נתונים ראוי לשמו, ניגע בכך רק במידה ההכרחית. גם מעט גישה לגרפיקה (או לגרפיקאי/ת) נדרשת, כדי לעצב לחצנים, רקעים, באנרים וכן הלאה. לסיכום: זוהי אמנות משולבת.

Visual InterDev הוא היום כלי הפיתוח הסטנדרטי לכתיבת אתרים מבוססי ASP. זהו כלי מדהים ביכולותיו, אך לא פשוט להפעלה. בצד הסבר מעמיק על השימוש הנכון בכלי, תקבלו גם עשרות טיפים וטכניקות עבודה מתקדמות שיחסכו לכם שעות רבות. ספר זה מיועד לאנשים הרוצים להכיר את הכלי ולשלוט בו ברמה מקצועית.



תרשים: ASP - אמנות משולבת.

Web Server

כאשר כתבנו קבצי HTML פשוטים (או לא), החיים היו קלים: הקשנו לחיצה כפולה על הקובץ שיצרנו, והדפדפן מייד הציג אותו. דפדפן, מעצם טבעו, יודע להציג קבצי HTML. מרגע שעברנו לתכנות ASP, אנו זקוקים לשרת web אמיתי, כזה שיודע לפענח את קבצי ASP ולייצר מהם קבצי HTML בזמן אמת. קיימים מספר שרתים כאלה, ונזכיר את שני החשובים ביותר:

IIS של מיקרוסופט, הוא חלק ממערכת ההפעלה NT. ליתר דיוק, חלק מהחבילה הנקראת NT Option Pack. זהו שרת אינטרנט חזק ביותר, המתאים לארגונים ולאתרים מסחריים. כאמור, IIS פועל בסביבת NT בלבד.

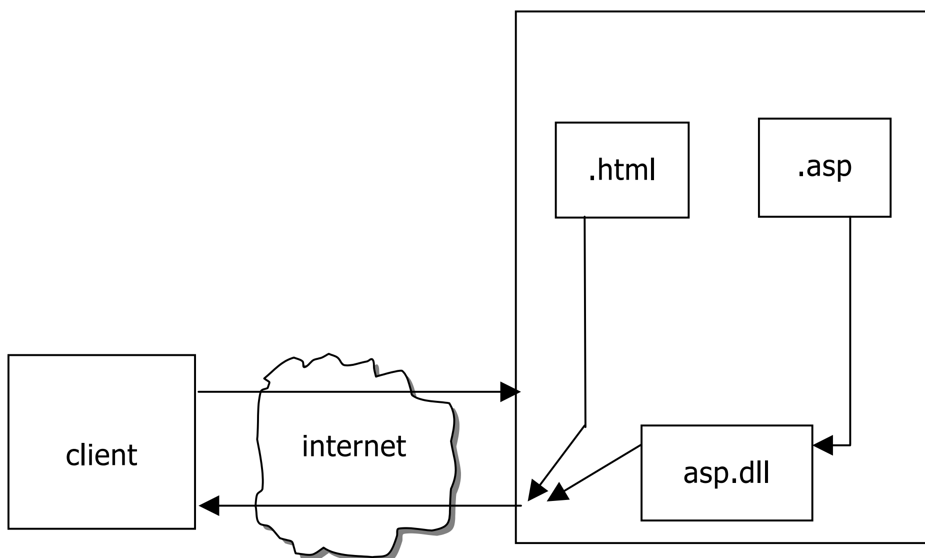
PWS - Personal Web Server, הוא מוצר עזר המיועד ללימוד ולאפליקציות פנים ארגוניות קטנות. מבחינת תכונות הוא דומה מאוד ל-IIS. הוא מעט פשוט יותר וקל יותר לניהול. יתרונו הגדול, הוא רץ על Windows 9x. קטן, ידידותי, ועושה את מלאכתו. אם אתם צריכים עוד סיבה לבחור בו, אז הנה: הוא גם מחולק חינם. PWS נמצא בתקליטור המצורף לספר זה, ניתן להוריד אותו מאינטרנט, והוא גם נמצא על דיסק ההתקנה של Windows 98. בפרק הבא נתקין יחד, בצורה מסודרת, את Personal Web Server.

אם מערכת ההפעלה שלכם היא **Windows 95**, תצטרכו לעדכן קודם את גרסת Winsock שברשותכם. את קובץ העדכון, **w95ws2setup.exe** תמצאו בתקליטור המצורף לספר זה, בתיקיה Software\Winsock. הריצו אותו תחילה, ורק אחר כך התקינו את PWS.

כיצד פועל שרת אינטרנט?

באופן בסיסי, תפקידו של שרת האינטרנט פשוט: לקבל בקשות מהגולשים ולהחזיר להם דפים. בניגוד למה שמקובל לחשוב, בעת גלישה באינטרנט אינכם "מחוברים" לאתר. כל הקשר שלכם איתו הוא דרך בקשות דפים. אחרי בקשה ומשלוח דף אין השרת יכול לדעת אם הגולש עדיין צופה בדף שלו, או עבר לאתר אחר. הוא יכול רק להניח זאת על פי קבוע זמן.

התנהגות השרת שונה בין בקשות לקבצי HTML לבין בקשות לקבצי ASP. כאשר לקוח מבקש קובץ בעל סיומת HTM או HTML, תפקיד השרת פשוט מאוד: לשלוח לו את הקובץ. כאשר הסיומת היא ASP השרת מתחיל לפענח את הקובץ שורה אחר שורה.



על עבודת הפיענוח אחראי הקובץ asp.dll. הוא עובר על שורות הקובץ: אם זו שורת HTML, הוא שולח אותה מייד ללקוח. אם זו שורת פקודה, הוא מבצע אותה. מה שמקבל הלקוח במקרה זה, הוא קובץ עם סיומת ASP, אך הוא אינו דומה לקובץ בשם זה שיושב בשרת, אלא הוא תוצאת הריצה. נראה זאת בדוגמאות בהמשך.

גרסאות שונות של ASP

יכולת ASP נובעת מקובץ יחיד בשם ASP.DLL, הנמצא בשרת האינטרנט. אתם יכולים למצוא קובץ זה במחשב שלכם, או בשרת. קובץ זה הוא שאחראי לפיענוח וביצוע של דפי ASP שנכתוב. הקובץ עבר מספר שינויים מאז הוצג לראשונה. כל שלב בהתפתחותו מקבל מספר גירסה. הגירסה המעודכנת ביותר בעת כתיבת שורות אלה, היא גירסה 3.

אם אתם משתמשים ב- Windows NT4, סביר להניח שיש לכם ASP בגירסה 2. התקינו את חבילת העדכון Service Pack4 ומעלה, שבין היתר גם מעדכנת את גרסת ASP ל-3. אם אתם משתמשים ב- Windows 2000 (Server או Professional), מותקן במחשבכם IIS 5.0, שכולל את ASP בגירסה 3.

כל הדוגמאות וההסברים בספר זה נכונים לשתי הגרסאות: 2 ו-3.

ASP במערכות שאינן מיקרוסופט

ASP הוא פיתוח של מיקרוסופט, לכן הסביבה הטבעית לעבודה היא סביבת מיקרוסופט: שרת NT או Windows 2000, וכלי הפיתוח Visual InterDev. אך ישנה אפשרות להריץ יישומי ASP גם על מערכות הפעלה אחרות. לדוגמה:

- Chili!ASP מאפשר להריץ יישומי ASP על שרת Netscape.

  פרטים נוספים: http://www.chilisoft.com

- Instant ASP של חברת Halcyon Software מאפשר להריץ ASP על מיגוון רחב של שרתים. ביניהם: Apache, Linux, Sun, Novell, ואפילו Windows NT.

  פרטים נוספים: http://www.halcyonsoft.com

גם בתחום כלי הפיתוח יש למיקרוסופט מתחרים, אך ללא ספק אף אחד מהם לא מגיע לעושר הכלים ומיגוון האפשרויות האדיר שמציע Visual InterDev.

בין כלי הפיתוח התומכים ב-ASP ניתן למנות את:

- HomeSite של חברת Allaire.

  פרטים נוספים: http://www.allaire.com

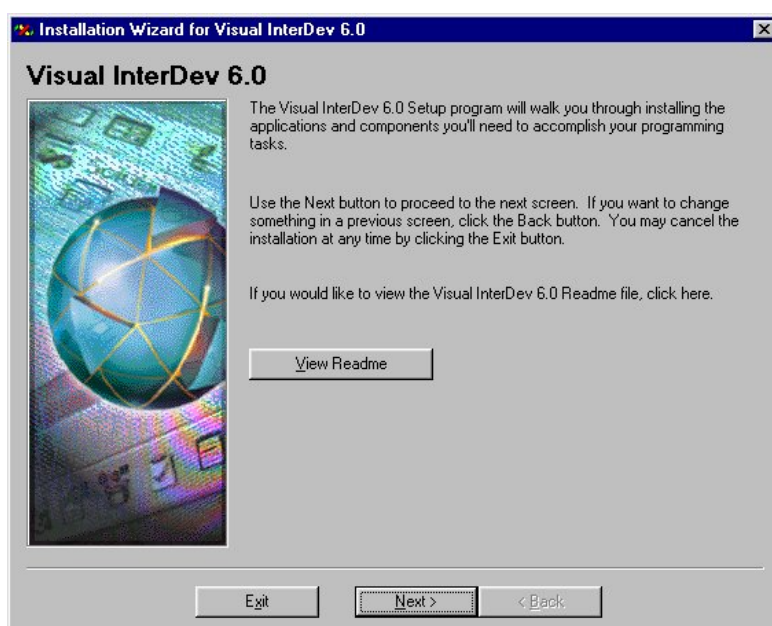
זהו אחד מעורכי ה-HTML הנוחים והקלים ביותר שנוצרו אי פעם. יתרונו, לעומת InterDev, בפשטות ההתקנה והעבודה. חסרונו: חלש יותר ואינו מציע את מיגוון האפשרויות האדיר של Visual InterDev.

את הגירסה האחרונה של HomeSite תוכלו למצוא בתקליטור המצורף לספר.

פרק 2

התקנת Visual InterDev

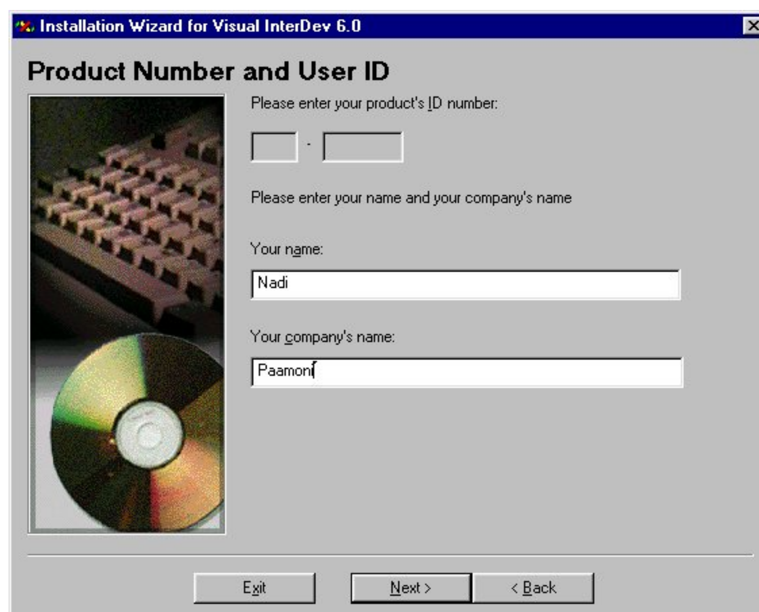
כשתכניסו את התקליטור המצורף לכונן, יופעל אוטומטית הדפדפן שלכם, ויופיע דף צבעוני עם תפריטים ואפשרויות רבות. כדאי לכם למצוא זמן ולטייל בתפריטים אלה, ידע ומידע רב מסתתרים מאחוריהם, דוגמאות, מאמרים, וגם לא מעט פרסומות של מיקרוסופט. בשלב זה, סגרו את הדפדפן, והפעילו את סייר החלונות. אנו עומדים להתקין את סביבת העבודה המשולבת: את כלי הפיתוח Visual InterDev, ואת השרת הביתי Personal Web Server. מומלץ להתקין את החבילה ב-Windows 98. אפשר גם להתקין ב-Windows 95, אך אז תצטרכו להתקין קודם גירסה מעודכנת של WinSock. היכנסו לתיקיה vi6trial ולחצו על הקובץ setup.



לחצו Next.



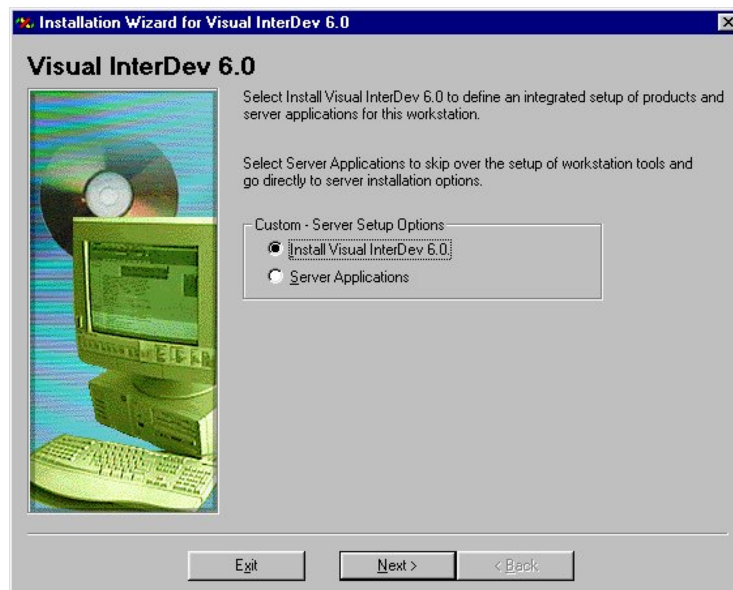
סמנו את I Accept the agreement.



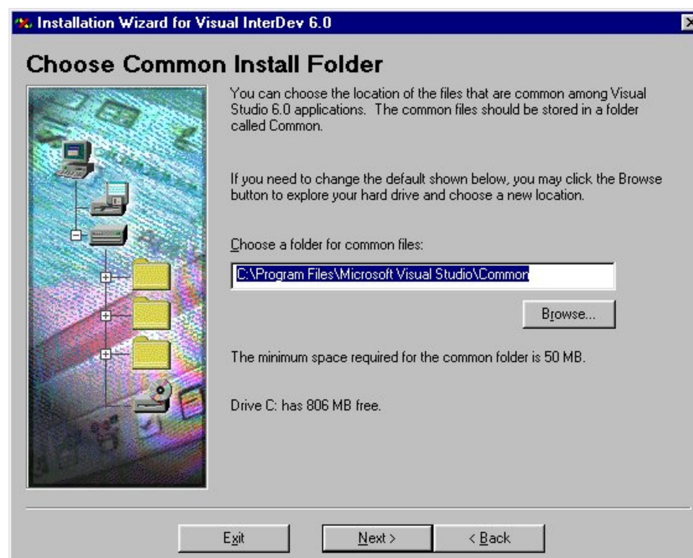
הקלידו שם כלשהו. אין צורך להקליד מספר סידורי.

אם מופיע חלון השואל אם להתקין Java Virtual Machine, ענו כן, ועברו הלאה.

מופיע חלון ההתקנה. בחרו Install Visual InterDev.

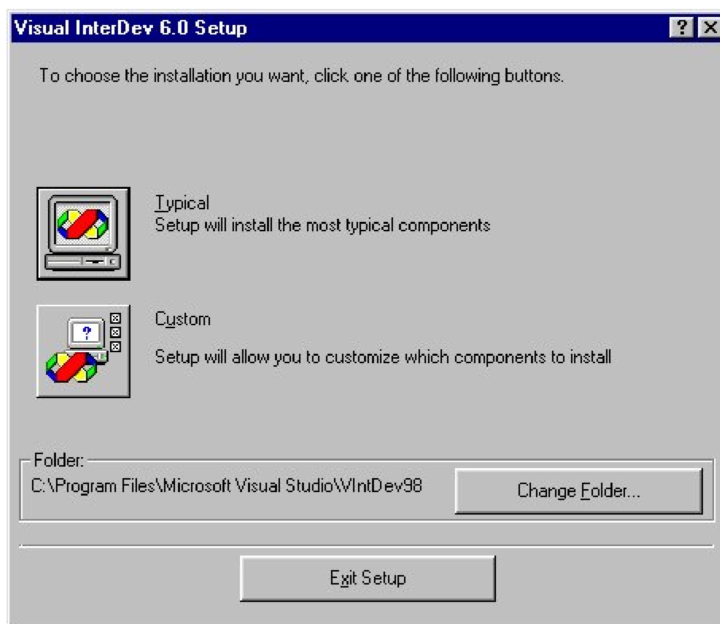


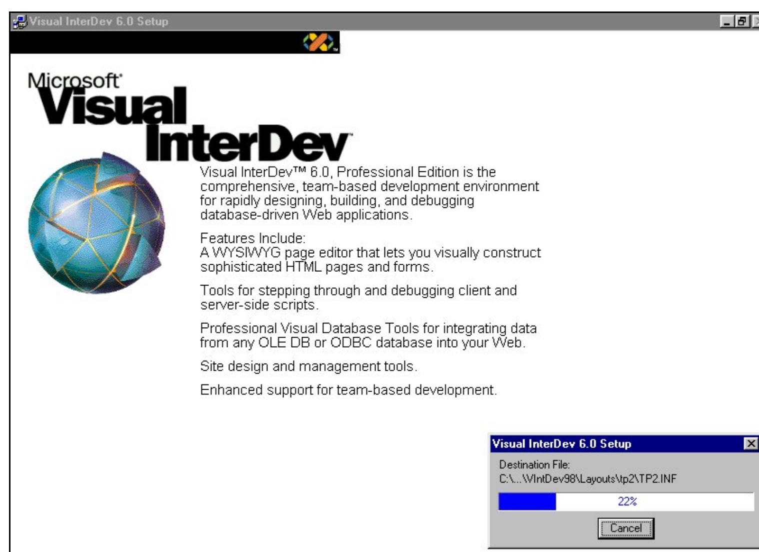
בחרו תיקיה להתקנה, או השאירו את ברירת המחדל (מומלץ).



עברו הלאה, עד שתגיעו לחלון בו יש לבחור את סוג ההתקנה.

בחרו בהתקנה רגילה - Typical.



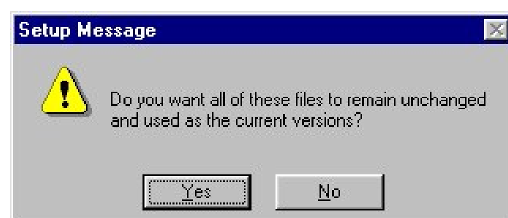


ההתקנה מתבצעת.

אם מופיע חלון השואל אם ברצונכם להחליף רכיבי DLL, ענו **No**.



לאחר מכן, כאשר תישאלו האם להשאיר את כל הרכיבים ללא שינוי, ענו **Yes**.



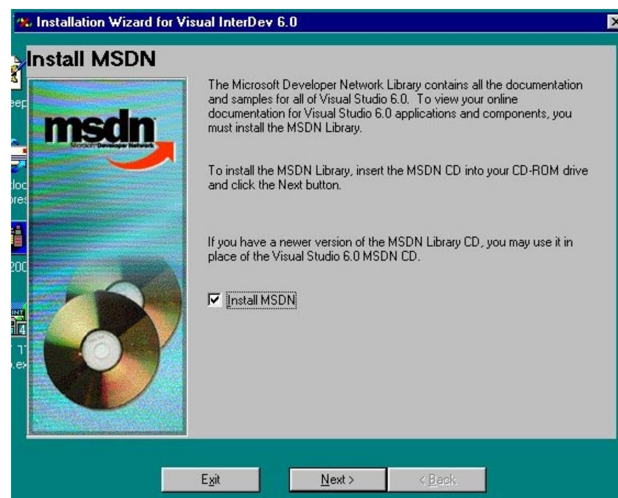
אם מופיעה תיבת דו-שיח "התנגשות בין גירסאות", בחרו תמיד לשמור את הקובץ הקיים (לחצו על **Yes**).



סיימתם חלק ראשון.

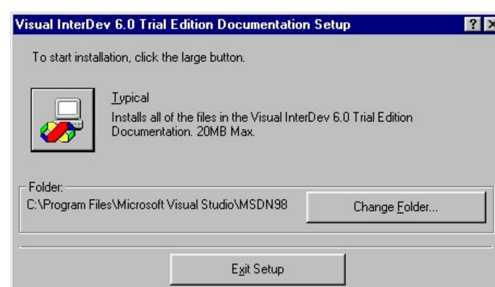
כעת, האם ברצונכם להתקין את MSDN? בוודאי. זהו בעצם קובץ העזרה של Visual InterDev.

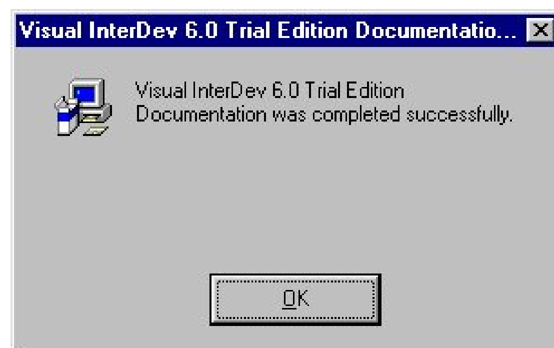
השאירו את התיבה Install MSDN מסומנת ולחצו Next.





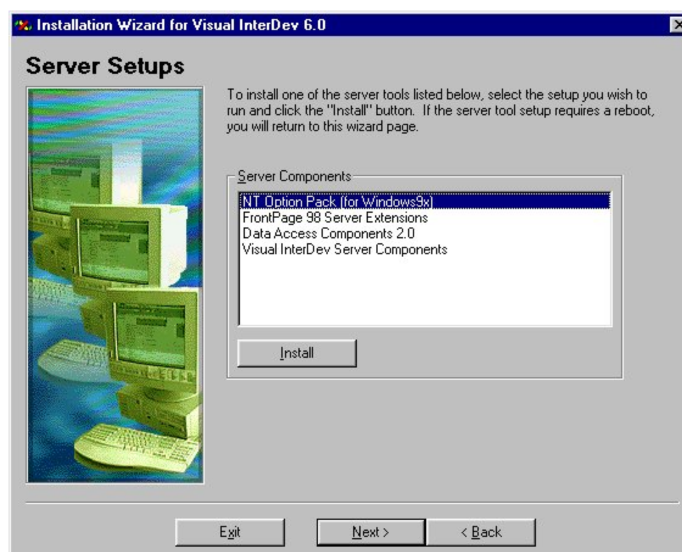
עברו לחלון הבא, ולחצו על לחצן התקנה הרגילה, Typical.





חלק התקנת ה"לקוח" עבר. כעת ניגש להתקנות השרת.

מוצגות ארבע אפשרויות. יש להתקין את כולן.



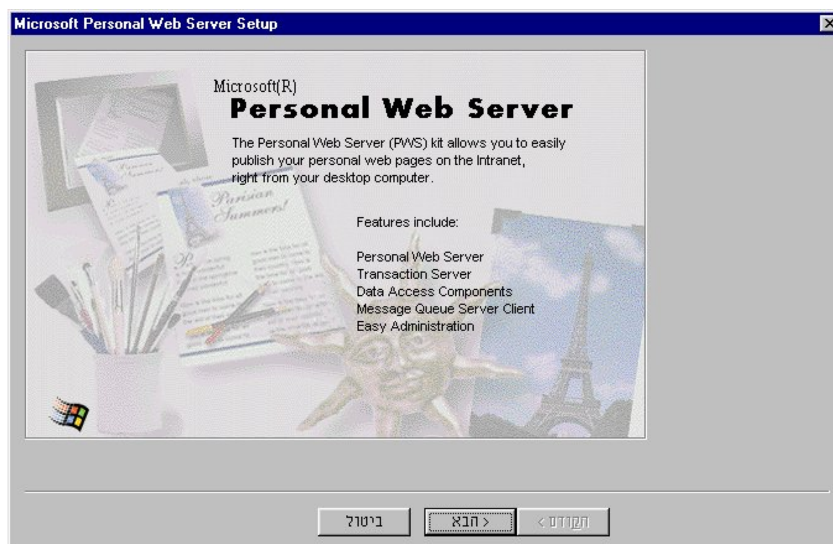
סמנו את הראשונה, NT Option Pack, ולחצו על Install. זה יגרום להתקנת PWS.

לחצו על Next להתקנה.

במחשבים בהם מותקן Windows 95, עלולה להופיע הודעת שגיאה זו:

WinSock 2 is Required to run this Setup

יש להתקין קודם גרסת WinSock מעודכנת. השאירו את תהליך ההתקנה פעיל, הפעילו סייר, היכנסו לתיקיה SoftWare\WinSock בתקליטור, והפעילו שם את הקובץ W95ws2setup.exe. לאחר מכן המשיכו בהתקנה.


Microsoft Personal Web Server Setup
Microsoft(R)
Personal Web Server
The Personal Web Server (PWS) kit allows you to easily
publish your personal web pages on the Intranet,
right from your desktop computer.
Features include:
Personal Web Server
Transaction Server
Data Access Components
Message Queue Server Client
Easy Administration
ביטול
הבא >
< הקודם


Microsoft Personal Web Server Setup
Microsoft
Personal Web Server
End User License Agreement
MICROSOFT WINDOWS NT 4.0 OPTION PACK
ADDENDUM TO END-USER LICENSE AGREEMENT FOR MICROSOFT SOFTWARE
IMPORTANT: READ CAREFULLY The Microsoft software components which are collectively identified above, including any online or electronic documentation (collectively referred to as Software Components), are subject to the terms and conditions of the End-User License Agreement for Microsoft Software (each such license referred to individually as a EULA), for the applicable Microsoft software product which you licensed previously from Microsoft, and the terms and conditions of this Addendum. By installing, copying or otherwise using the Software Components, you
Accept
Decline
ביטול
הבא >
< הקודם

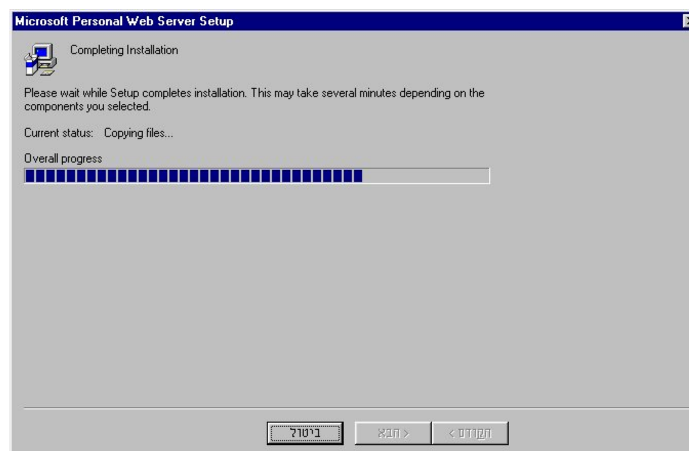
לחצו Accept.



בחרו בהתקנה רגילה (Typical).

קבלו את ברירת המחדל למקום ההתקנה של השרת.

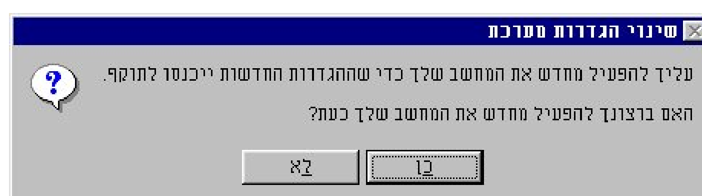




שתו קפה בעת התקנת Personal Web Server.

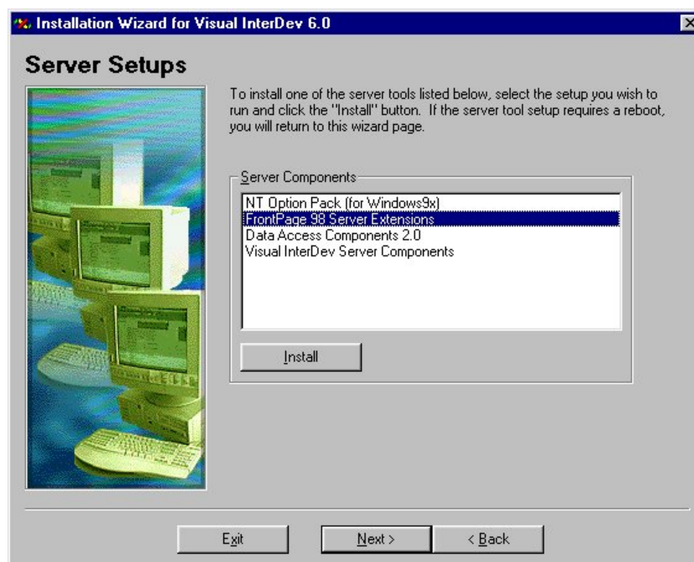


הצלחתם!



האם ברצונכם להפעיל מחדש? כן, איזו שאלה!

עבור לרכיב הבא. סמן FrontPage Server Extensions, ולחצו Install.

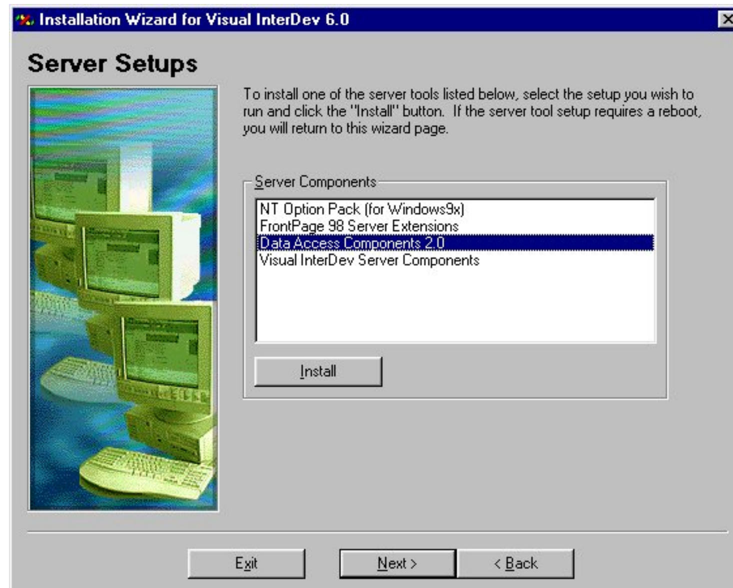


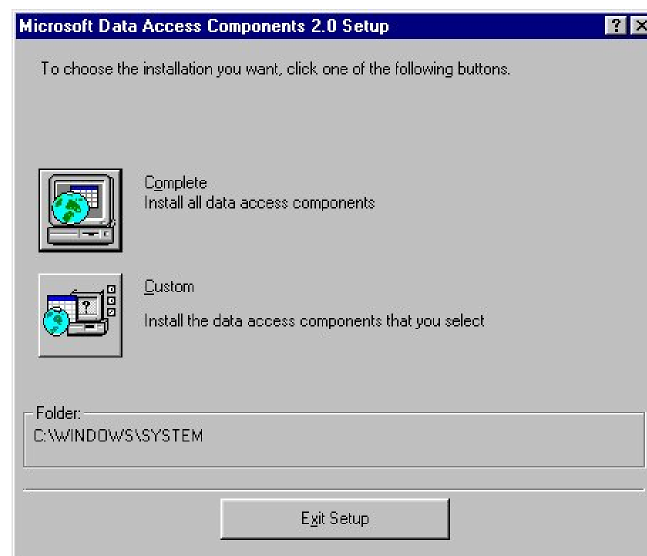


לחצו Next ובצעו את ההתקנה.

חלק שלישי של התקנת השרת : רכיבי גישה לנתונים.

בחרו Data Access Components 2.0 ולחצו Install.



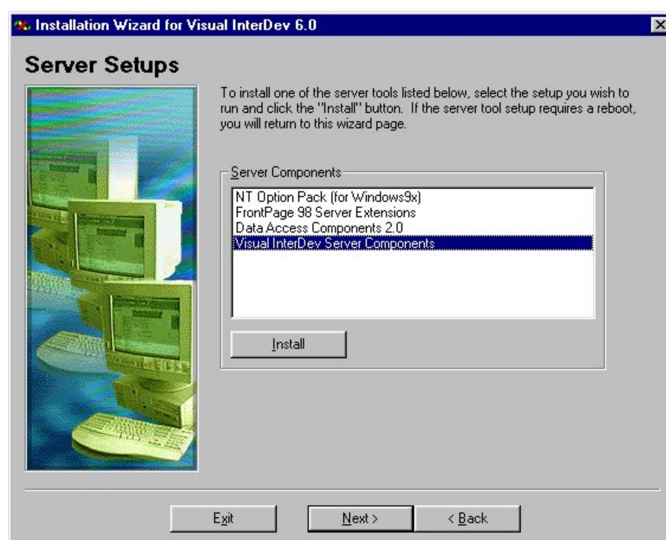


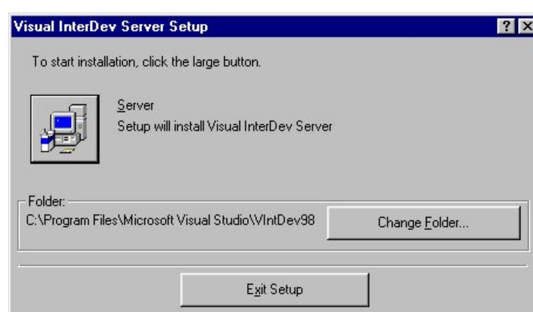
לחצו Complete להתקנת רכיבי הגישה לנתונים.



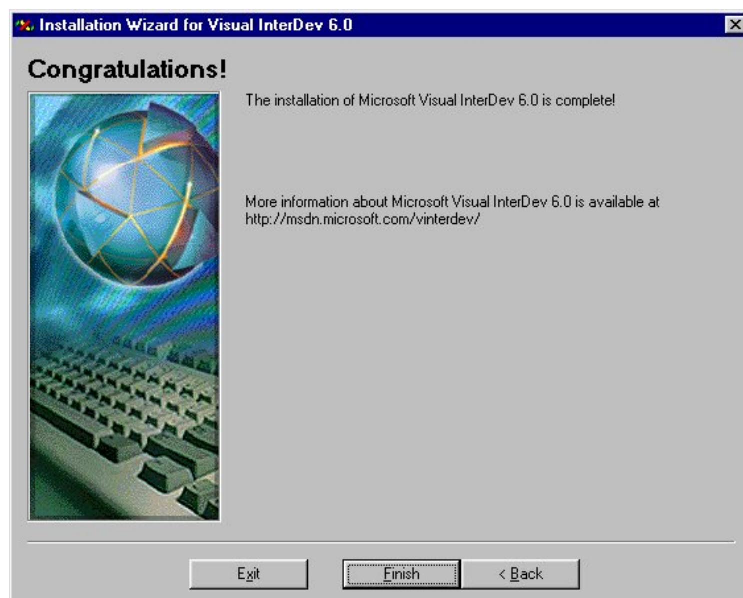
יופי, שלב נוסף הסתיים.

שלב אחרון: התקנת רכיבי השרת. סמנו Visual InterDev Server Components ולחצו Install.





לחצו על לחצן Server.



נהדר! Visual InterDev, ושרת האינטרנט Personal Web Server מותקנים במחשבים שלכם.

כעת הביטו אל שורת היישומים במחשב שלכם, ותראו את הסמל של PWS (הימני ביותר, ליד השעון).

כאשר הוא כבוי הוא נראה כך :



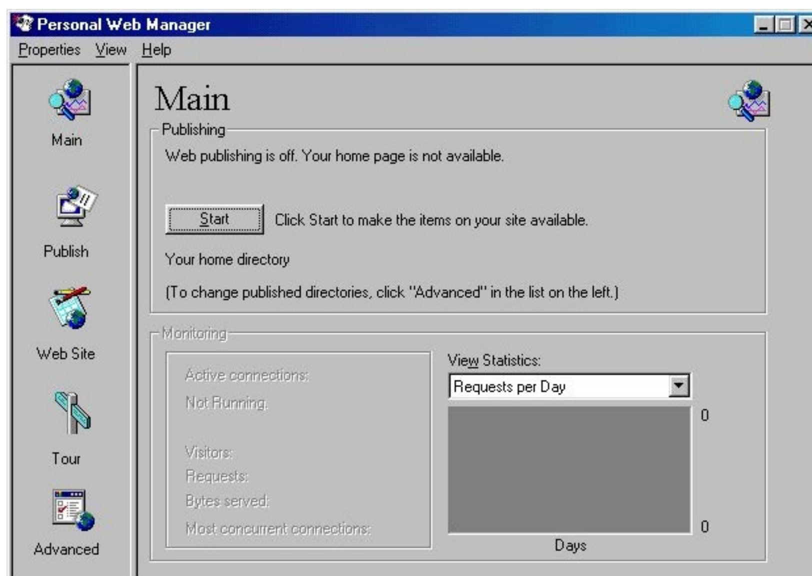
וכאשר הוא מופעל הוא נראה כך :



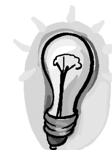
אם הוא כבוי, וברצונכם להפעיל אותו, לחצו פעמיים על הסמל (Icon), ותקבלו את מסך הניהול של Personal Web Server.

אם הסמל לא מופיע כלל, לחצו על :

התחל < תוכניות < Microsoft Personal Web Server < Personal Web Manager.



לחצו על לחצן Start והשרת מופעל.



טיפ:

PWS הוא יישום גוזל משאבים! כאשר אינכם עובדים בפיתוח לאינטרנט, כדאי לסגור אותו ולקבל ביצועי מחשב טובים בהרבה.

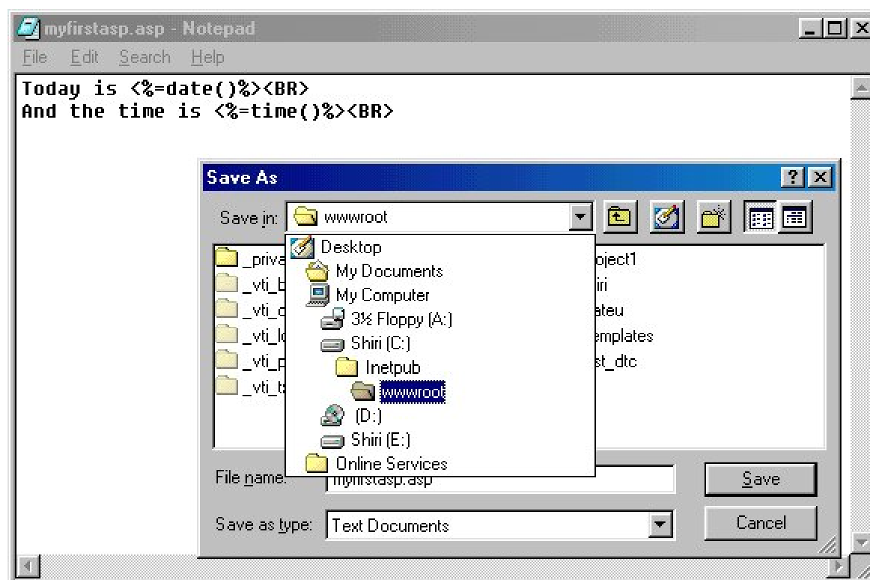
בדיקת תקינות

כעת נוודא שהשרת שלנו אכן פועל. נכתוב דף ASP בסיסי, ונריץ אותו.

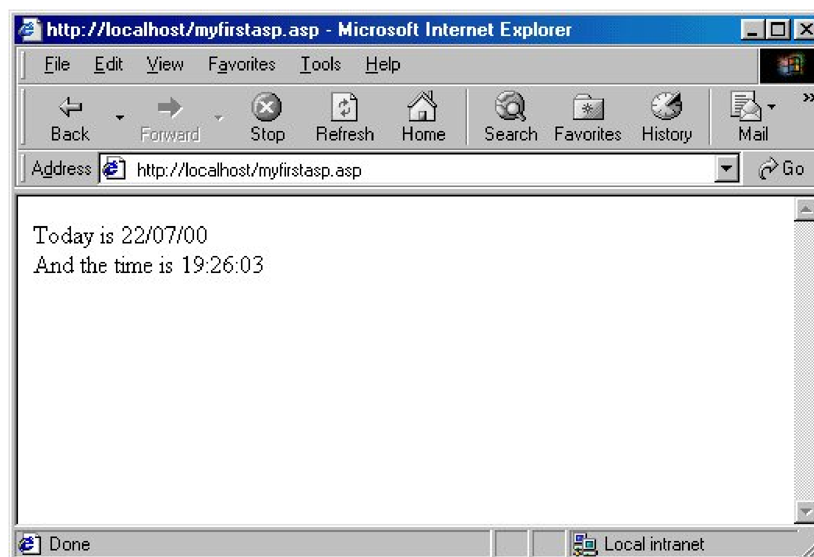
הפעילו את פנקס הרשימות, הידוע בשמו NotePad, וכתבו בו כך :

```
Today is <%=date()%><BR>
And the time is <%=time()%><BR>
```

שמרו את הקובץ בשם myfirstasp.asp בתיקיה c:\inetpub\wwwroot. זו התיקיה בה מצפה PWS למצוא את קבצי ASP שלכם.



כעת הפעילו את הדפדפן שלכם, ובשורת הכתובת כתבו: **localhost/myfirstasp.asp** והקישו Enter. אם הכל בסדר, תראו משהו דומה לזה:



זה פועל!

המילה localhost וכן הכתובת 127.0.0.1 הן כתובות שמורות, ומשמעותן: כתובת המחשב הנוכחי.

במילים אחרות, היית יכול לכתוב בשורת הכתובת: 127.0.0.1/myfirstasp.asp

ניתן לפתח פרויקטי ASP בשיטה זו של כתיבה ב- NotePad, אך זוהי שיטה מייגעת ומאוד קשה לתחזוקה. את יתרת הספר נבלה במחיצת Microsoft Visual InterDev, כלי המיועד לכתיבת פרויקטים ב-ASP, שיעשה לנו חיים קלים ויסיר מעלינו דאגות רבות. אך חשוב להבין ולזכור, כי דפי ASP בסופו של דבר הינם דפי טקסט רגילים שכל עורך טקסט פשוט (כדוגמת notepad) יכול ליצור ולערוך.

מה לעשות אם PWS לא עובד?

אם מסיבה כלשהי PWS לא עובד, בצעו את הפעולות הבאות:

ראשית, בדקו דברים טריוויאליים: האם המוצר הותקן במלואו? האם המוצר מופעל? ודאו ש- PWS אינו כבוי בטעות.

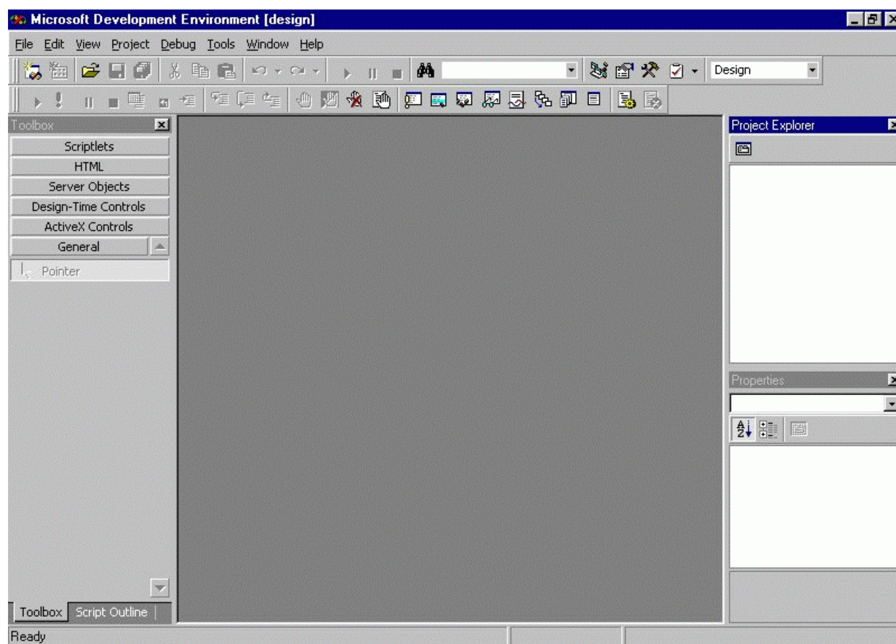
שנית, קראו את תיעוד התוכנה. בחרו Start > Programs > Product Documentation. כשייפתח המסמך, חפשו את הדף TroubleShooting. הוא מכיל עצות חשובות לפתרון בעיות ב- PWS. במיוחד שימו לב להבדל בין הפעלת PWS ברשת מקומית לבין הפעלתו במחשב בודד.

פרק 3

יצירת פרויקט ראשון ב- Visual InterDev

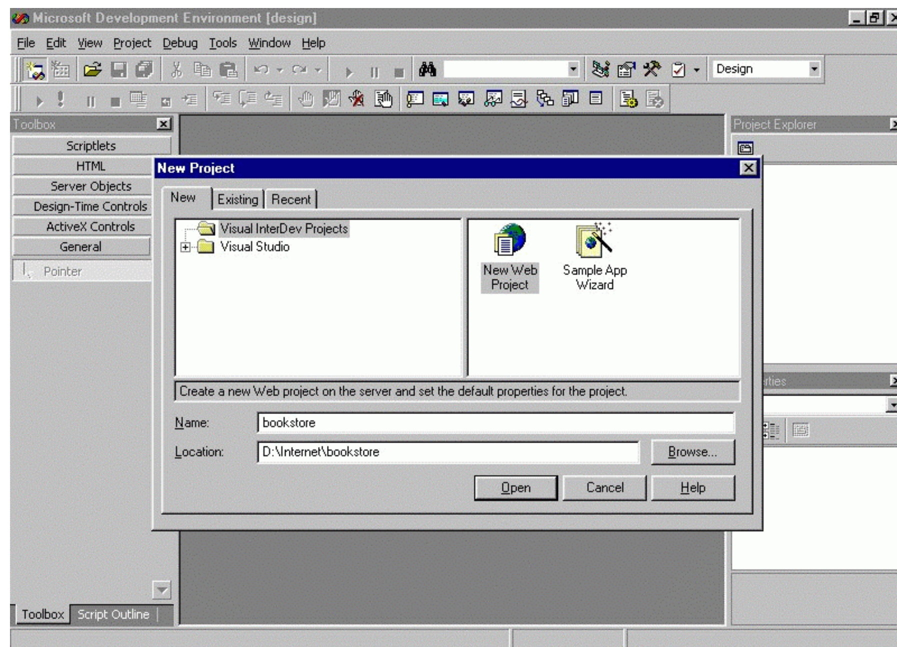
קובץ ASP, כאמור, אינו אלא קובץ טקסט רגיל, וניתן לערוך אותו בעזרת כל עורך טקסטים, כגון Notepad. אולם, כלי הפיתוח Visual InterDev 6 של מיקרוסופט, מקל מאוד על העבודה ומספק סביבת עבודה נוחה ותומכת. רוב הדוגמאות בספר זה יעבדו היטב גם ללא VID אך מומלץ להתקין אותו וללמוד להשתמש בו. תחילה, נכיר את סביבת העבודה, ונפעיל את Visual InterDev בפעם הראשונה.

לחצו על לחצן **התחל**, בחרו **תוכניות** < **Microsoft Visual Studio 6.0** < **Microsoft Visual InterDev 6.0**.

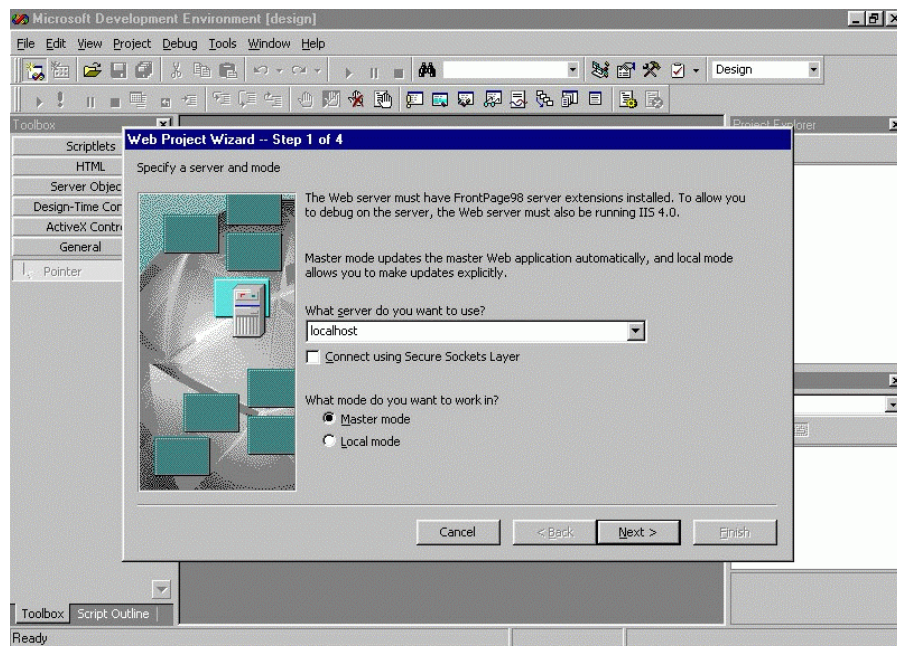


בחלק העליון - תפריט ראשי וסרגל כלים כמו בכל יישום. בצדדים, ולעיתים גם בחלק התחתון, מופיעים חלונות עזר. חלונות אלה ניידים : ניתן להזיזם ממקומם, לשנות את גודלם לפי הצורך, להחליף מקומות ביניהם, להסתיר ולהראות אותם לפי הצורך. אם החלון שלכם נראה מעט שונה - לא נורא.

נתחיל מייד בפרויקט שלנו, שיהיה כמובן, חנות ספרים באינטרנט. אם לא עלה חלון פרויקט חדש מעצמו, העלו אותו על ידי בחירת File < New Project.



שימו לב לכך, שחלון New Project שונה מחלונות יצירה בתוכנות אחרות. יש בו אפשרויות רבות, כמו פתיחת פרויקטים קיימים, פרויקטים אחרונים שטופלו ועוד. תחילה, בתיבה location הקלידו \:c, או בחרו בתיקיה בה תרכזו את כל פרויקטי האינטרנט שלכם. הקלידו את שם הפרויקט : bookstore בתיבה Name. ודאו שמסומן New Web Project, ולחצו על Open.



אשף יצירת הפרויקטים מתחיל לעבוד, והוא בשלב 1. שאלתו היא: מיהו שרת האינטרנט מולו נעבוד? (What server do you want to use?) אם השרת שלכם מותקן במחשב בו הינכם עובדים, הקלידו localhost, או את הכתובת 127.0.0.1. המשמעות זהה - השרת המותקן במחשב המקומי. אם זהו שרת ארגוני המותקן במחשב נפרד, הקלידו את שמו או את כתובת ה-IP שלו.

שימו לב לתיבת הסימון Connect using Secure Sockets Layer הקובעת אם להשתמש בפרוטוקול SSL. זהו פרוטוקול הצפנה שפותח על ידי Netscape, ומשמש לאבטחת תעבורת http משרת האינטרנט. מידע נוסף על SSL, בפרק מיוחד לכך. בשלב זה השאירו את התיבה לא מסומנת.

החלטה נוספת שעומדת בפנינו היא אופן העבודה: master או local. הסבר קצר: מצב local פירושו בידוד בין עבודתכם בתחנה המקומית לבין שרת האינטרנט ה"חי". רק כאשר תחליטו שקובץ מסוים מוכן לעלות לאוויר, תעלו אותו. עד אז הוא נמצא רק בתחנת העבודה שלכם, ורק אתם יכולים לראות אותו.

מצב master הוא מצב של אחידות בין תחנת העבודה שלכם לבין השרת. כל שינוי שתבצעו בקובץ, מופיע מייד באתר. ליתר דיוק, פעולת השמירה גורמת לעדכון האתר. זה נוח ללימוד ולעבודת פיתוח מול שרת מקומי כדוגמת PWS, אך לא מעשי לעבודה מול שרת חי או בעבודת צוות.

יש להבין נקודה מרכזית בעבודה עם Visual InterDev: הפרויקט שלכם **קיים פעמיים**. הוא פשוט משוכפל. סט קבצים אחד נמצא בתיקיה בו בחרתם להקים את הפרויקט, וסט שני בשרת האינטרנט שלכם. אבל אתם אומרים "שרת האינטרנט שלי הוא PWS

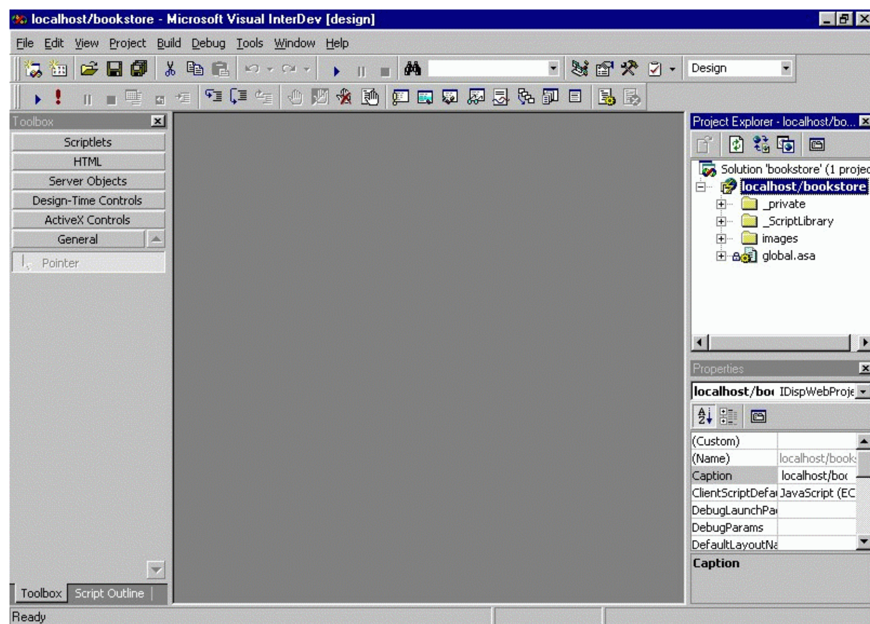
מקומי!" נכון, ושם בדיוק נמצא העותק השני. לאחר יצירת הפרויקט, הציצו בעזרת הסייר ב- c:\inetpub\wwwroot ושם תראו תיקיה חדשה בשם bookstore, שמכילה בדיוק את אותם קבצים ותיקיות שמכיל הפרויקט שלכם!

העבודה במצב local חשובה מאוד בעת עבודה בצוות. נניח שאתם עובדים בצוות, והבוס מבקש מכם לעשות שינויים מהפכניים בדף ההזמנה. שינויים אלה יקחו לכם בערך יום עבודה. בזמן זה, חבריכם לעבודה עובדים על דפים אחרים, אך משתמשים בדף ההזמנה כחלק מהתהליך. אינכם רוצים להפריע להם. ומה עם גולשי האינטרנט? האם אתם מוכנים שהם יראו את הדף שלכם בזמן העבודה? אולי נתלה שלט גדול – "סגור לרגל שיפוצים"? יהיה לכם נוח ביותר, אם תוכלו לעבוד ללא הפרעה על דף ההזמנה, בעותק מקומי, וכולם ימשיכו לראות את העותק הישן. כשתהיו מוכנים, עדכנו את השרת בדף החדש שלכם. גם חבריכם לעבודה וגם גולשי האינטרנט יראו לפתע את הדף החדש. זה בדיוק מה שקורה במצב local.

השאירו את הסימון בברירת המחדל - Master ולחצו Next.

שני המסכים הבאים עוסקים בעיצוב, סגנון, בחירת רקעים, ושאר ענייני קוסמטיקה. לעת עתה נתעלם מהם. בפרק מיוחד לכך, נעסוק בסגנונות ועיצובים. לחצו Finish לסיום.

המחשב יעבוד מספר שניות ובסיומן המסך ייראה דומה לזה:



יופי! יצרתם את פרויקט Visual InterDev הראשון שלכם. הבה נכיר את סוגי הקבצים המרכיבים את הפרויקט.

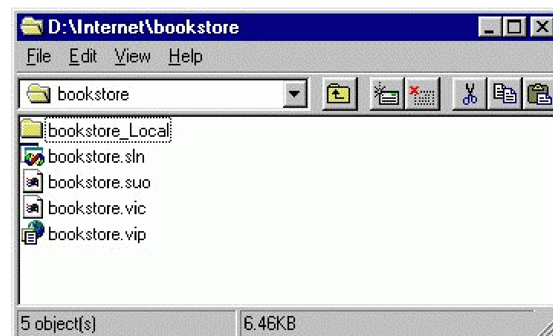
מכיון שבחרנו לעבוד במצב Master, לא נצטרך לעסוק הרבה בשאלת עדכון השרת. פעולה זו תתבצע אוטומטית. אך חשוב בכל זאת לזכור, שהקבצים שרצים בפועל, הם אלה שבשרת, ולא אלה שבעותק ה"מקומי" (מקומי במרכאות, מפני שאצלנו הכל בעצם, מקומי).

דוגמה לחשיבות ההבנה של מי באמת רץ: נניח שאתם רוצים להוסיף תמונה כלשהי לאתר. שמרתם את התמונה בדיסק שלכם, בתיקיה images, כמו שצריך. עשיתם אליה הפניה מתוך הדף, והיא מסרבת להופיע בעקשנות. הסיבה היא שהשרת לא יודע עליה. התמונה החדשה אינה כלולה בפרויקט, ולכן מנגנון העדכון האוטומטי לא דואג להעתיק אותה לשרת. יש לצרף את התמונה לפרויקט ולבקש עדכון שרת.

כעת נתבונן בקבצים שנוצרו. תחילה, אתם שואלים, היכן הם נוצרו? ובכן, זה קל מאוד. ראשית נחפש את העותק המקומי, זה שבו נעסוק ואותו נעדכן. הוא נוצר, אם לא ביקשתם אחרת, תחת התיקיה InterDev בדיסק שלכם. לא זוכרים? לחצו לחיצה ימנית על שם הפרויקט ב- Project Explorer (באות עבה כתוב localhost/bookstore) ובחרו properties. שם, תחת project, כתוב בדיוק היכן נמצאים קבצי הפרויקט.

עותק השרת נמצא, כמה מפתיע, בשרת. אם אתם עובדים עם PWS, יימצאו הקבצים בדרך כלל ב- c:\inetpub\wwwroot\bookstore.

נתבונן בעותק המקומי, בתיקיה bookstore:



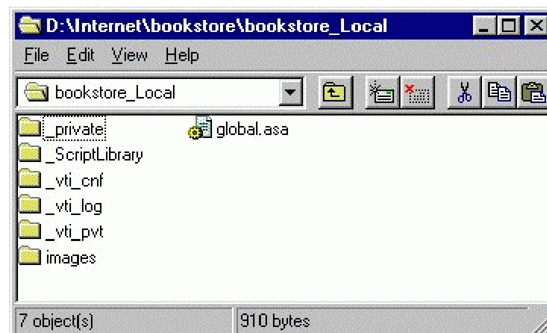
כאן נמצאים קבצי ניהול הפרויקט, ובתוך התיקיה bookstore_Local נמצאים הקבצים עצמם.

bookstore.vip - קובץ הפרויקט שזה עתה יצרנו. הקובץ מכיל מידע על שמות ומיקומי הקבצים המשתתפים בפרויקט, מיקום השרת וכן הלאה.

למרות שזהו קובץ ASCII רגיל, לעולם אל תערכו אותו ישירות, אלא רק דרך ממשק התוכנה.

bookstore.sln - קובץ הפתרון (solution). פתרון הוא צירוף של מספר פרויקטים יחד. לעת עתה אנו עוסקים בפרויקט בודד, ולכן אין כרגע לקובץ זה חשיבות.

ניכנס לתוך התיקיה bookstore_Local, ונראה את התמונה הבאה:



זו התיקיה שבה יימצאו הקבצים שניצור: קבצי HTM, ASP וכן הלאה. עבור תמונות וגרפיקה נוצרת עבורנו תיקיה מיוחדת - images, אך מותר לנו בהחלט ליצור תיקיות חדשות עבור לחצנים, קטעי וידאו, וכל מה שנרצה.

שימו לב לתיקיה ScriptLibrary. היא מכילה עשרות קבצי ASP ו-HTML, שישמשו אותנו בעבודתנו ויחסכו לנו הרבה זמן ואנרגיה.

הקובץ global.asa הוא קובץ ייחודי בפרויקט, ובו נכתבות התוכניות שרצות בתחילת ובסיום היישום, וכן בתחילה ובסיום של כל session. הסברים מפורטים בהמשך.

נניח לרגע את הפרויקט שלנו, ונכיר מעט את סביבת העבודה.

חלונות המשנה
ב- Visual InterDev

בתוך מסגרת החלון של Visual InterDev צפים להם מספר חלונות עזר. החלונות ניתנים להזזה, לשינוי גודל ולהצמדה זה לזה (Docking). נסביר את תפקידיהם:

חלון סייר הפרויקט - Project Explorer



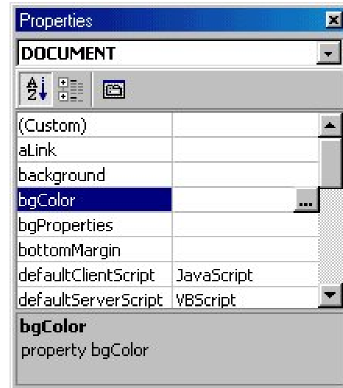
אחד מהחלונות החשובים ביותר ב- Visual InterDev. יש לו שני תפקידים:

1. לייצג את כל הקבצים המשתתפים ביישום, ולהציג את מצבם כרגע ביחס לעותק המרכזי שבשרת.
2. זהו כלי לניהול קבצים בעת עבודה בצוות. כאן תוכלו להוסיף קבצים לפרויקט, למחוק אותם, להביא קבצים מהשרת או לשלוח אליו.

שימו לב לסמלים השונים שיש לקבצים. כל אחד מהם הוא בעל משמעות. אם הסמל אפור– Disabled, תצטרכו להביא עותק מהשרת לפני שתוכלו לעבוד עליו (לחיצה ימנית על הקובץ, ולבחור Get Working Copy).

כאשר אתם עובדים על קובץ, אחרים לא יוכלו לעבוד עליו. כדי לשחרר אותו, לחצו לחיצה ימנית ובחרו Release Working Copy. אם יש מנעול ליד הסמל, זה סימן שיש עותק מקומי בלבד של הקובץ. עלול להיות הבדל בין העותק המקומי לעותק הראשי, אם ביצעתם שינויים מאז העדכון האחרון, או אם איש פיתוח אחר ביצע שינויים בקובץ הראשי. כאשר עובדים לבד בשרת מקומי, כל זה לא חשוב.

חלון תכונות - Properties

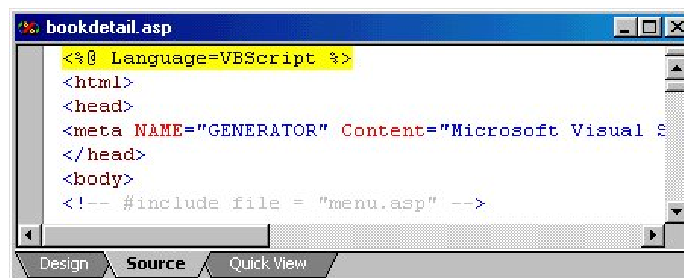


בדומה לשאר כלי מיקרוסופט, חלון Properties הוא כלי מרכזי ובו, כמה מדהים, קובעים את תכונות האובייקטים השונים. לדוגמה, אם נלחץ על הלחצן שליד התכונה bgColor, נקבל חלון בחירת צבעים ונוכל לקבוע את צבע רקע הדף שלנו.

חלון עריכה

בחלון העריכה נכתוב את דפי הפרויקט. זהו עורך חזק ביותר, המורכב משלושה חלונות. ניתן בכל עת לעבור ביניהם. אלו שלוש נקודות מבט על אותו קובץ. כל שינוי שנבצע באחד מהם ייראה גם באחרים.

Source



מצב עריכת טקסט. דומה להרבה מאוד עורכי HTML, אבל עם תכונות רבות כח: זיהוי מילים שמורות, קוד צבעוני לקבוצות מילים – פקודות, קבועים, תגי HTML וכו'. תכונה חשובה שנקראת IntelliSense - העורך "מכיר" את מבנה האובייקטים של ASP ועוזר לנו לכתוב. אתם כותבים את תחילת המילה - והוא משלים. אתם כותבים אובייקט – ומייד מוצגת בפניכם רשימת תכונות ופעולות שלו. נחמד, יעיל, וחוסך טעויות הקלדה.

Design

מצב עיצוב. במצב זה העורך מנסה להתחפש ל- FrontPage. ניתן לכתוב בגופנים שונים, לצייר קווים, לצבוע וכן הלאה - הכל ב- WYSIWYG, דומה מאוד למציאות. אך זהו מסך עיצוב בלבד ופקודות תכנות שכתבתם, מכל סוג, לא יבוצעו.

Quick View

מצב "מבט מהיר" דומה מאוד למצב Design, אך בהבדל אחד: הוא מפעיל Client Side Scripts. כלומר, אתם תראו את הדף כפי שהוא ייראה לגולש בדפדפן, כולל תפקוד של תוכניות צד-לקוח.

לדוגמה, ניקח פקודת VBScript פשוטה, המציגה את התאריך של היום. נכתוב אותה במצב Source:

```
ASP Page1.asp
<HTML>
<HEAD>
<BODY>
<B>Today is
<SCRIPT LANGUAGE=VBScript>
  document.write date()
</SCRIPT>
</B>
</BODY>
</HTML>
Design | Source | Quick View
```

זוהי דוגמה לתכנות צד לקוח. כלומר, אין צורך בשרת WEB כדי להריץ את הקוד. הדפדפן יודע לעשות זאת בעצמו.

נעבור למצב **Quick View** כדי לראות את התוצאה:

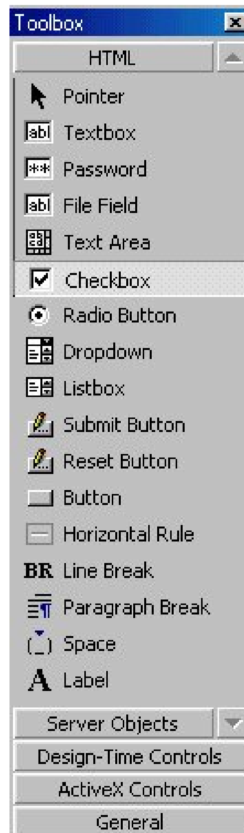




הערה:

שני המצבים הגרפיים, Design ו- Quick View, אינם יודעים להריץ פקודות ASP. לשם כך דרוש שרת אינטרנט. ככל שהדף מכיל יותר פקודות Server Side, הולכים מצבים אלה ונעשים פחות שימושיים. אז נלחץ לחיצה ימנית ונבחר View in Browser. בשלב העיצוב הראשוני, הם מאוד נוחים.

ארגז כלים - ToolBox



ארגז הכלים, כמו אצל כל מוסכניק, הוא המקום בו מרוכזים כלי העבודה. במקרה שלנו, אלה לא רק כלי עבודה, אלא גם אבני בניין. ארגז הכלים מחולק למחלקות: HTML, Server Objects ועוד. לחיצה על כל אחת מהמחלקות מראה את תכולתה. ניתן לגרור רכיבים מארגז הכלים לחלון העריכה.

כדי להכיר את ארגז הכלים, פתחו דף חדש, וגררו אליו כמה מהרכיבים המופיעים במחלקת HTML. ניתן לגרור אל חלון העריכה, כשהוא במצב Source או במצב Design. נסו זאת, וראו את התוצאה.

רשימת משימות - Task List

לא במיוחד שימושי בשלב זה. סגרו אותו, וכך יהיה לכם יותר מקום פנוי על המסך.



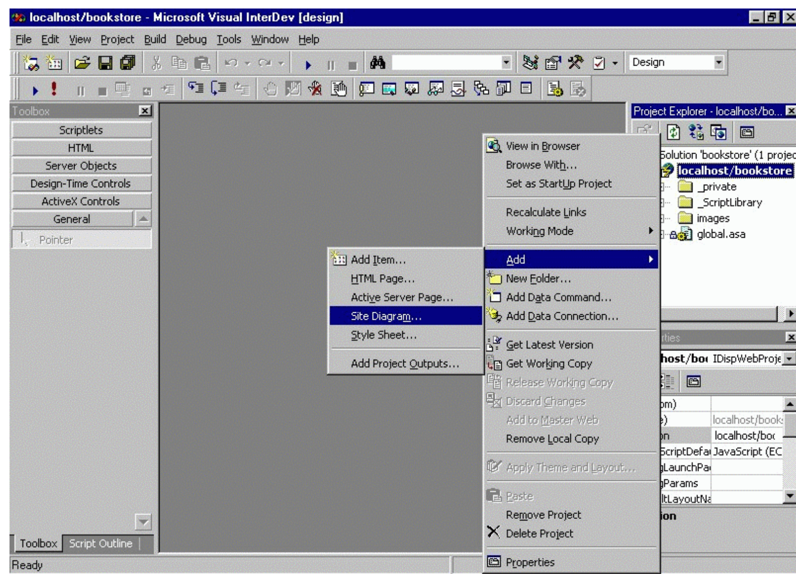
טיפ:

חלונות המשנה בתוך סביבת Visual InterDev ניידים. אפשר להעביר אותם ממקום למקום, לסגור אותם ולפתוח כשצריך. אבל אם הפרדתם אותם זה מזה, קצת קשה לעגן אותם חזרה למקומם, פעולה שנקראת Docking. הנה שני טיפים שיעזרו בכך:

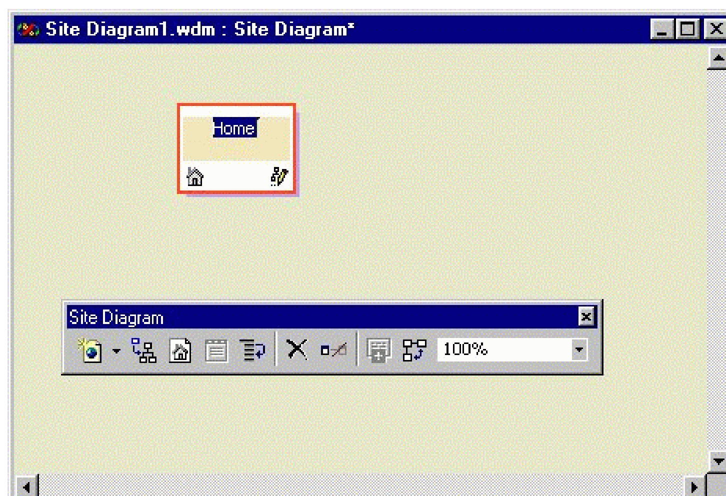
1. לחיצה על מקש Ctrl בזמן הזזת חלון, מונעת Docking.
2. כדי לעגן חלון אחד בתוך חלון שני, תפסו את החלון הראשון בכותרתו וגררו אותו לכיוון החלון השני, כאשר אתם מקרבים את **העכבר** אל **כותרת** החלון השני.

יצירת תרשים האתר

שלב ראשון בבנייה של אתר אינטרנט, בעצם בבנייה של כל דבר, הוא תכנון. ככל שתשקיעו יותר מחשבה בתכנון, כך תצטרכו פחות לשנות ולתקן בהמשך. Visual InterDev מספק כלי נפלא לתכנון האתר והרבה יותר מכך. זהו כלי לתחזוקה, להתמצאות ולתיעוד. בחלון Project Explorer לחצו לחיצה ימנית על שם הפרויקט, בחרו Add. ייפתח תפריט נוסף (שימו לב לכל האובייקטים שניתן להוסיף). בחרו Site Diagram.



אתם מתבקשים לתת שם לתרשים האתר, השם שניתן כברירת מחדל, Site Diagram1 הוא בסדר. כל שם אחר שתרצו יתקבל.



זהו התרשים המתקבל, בו מוכן עבורנו דף ראשון בשם Home. ההצעה מתקבלת בברכה, שכן אנו באמת זקוקים לדף ראשי שיהיה מעין נקודת זינוק וקישור ליתר הדפים. שימו לב לסרגל הכלים שהופיע. העבירו את העכבר על הסמלים וראו מה כל אחד עושה.

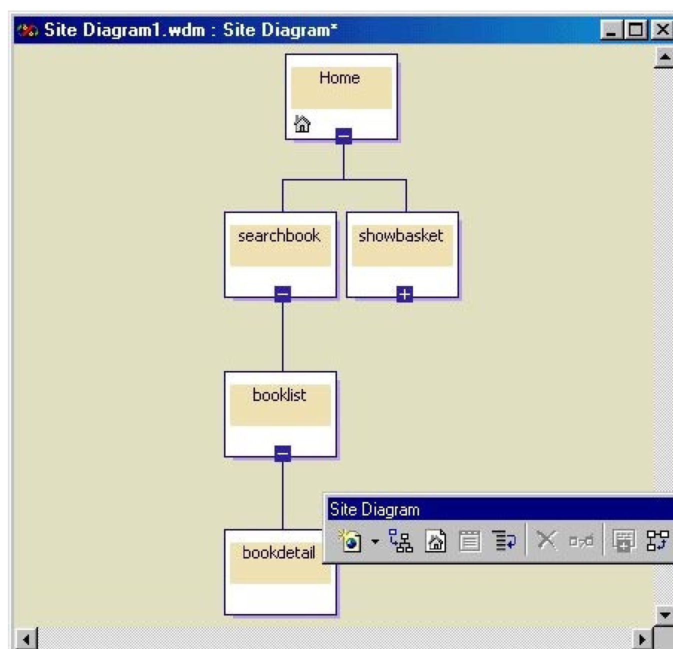
האתר שלנו יהיה חנות ספרים וירטואלית, לכן דרושים לנו בשלב הראשון הדפים הבאים:

שם הדף	קובץ	תיאור
דף בית	home.html	דף פתיחה, נקודת מוצא ליתר הדפים.
חיפוש	Searchbook.asp	דף המאפשר לגולש לחפש ספר על פי מילת מפתח.
רשימת ספרים	booklist.asp	רשימת ספרים. זוהי תוצאת החיפוש. הגולש הקליד מילה לחיפוש והוא מקבל את רשימת כל הספרים ששמם מכיל מילה זו.
פרטי ספר	Bookdetail.asp	דף פרטים מלאים על הספר, כולל מחיר, תמונה, חוות דעת גולשים, וכמובן, לחצן קנייה. לדף זה מגיע הגולש מתוך רשימת תוצאות החיפוש.
הוספת ספר	Showbasket.asp	דף המראה את מצב סל הקניות של הגולש.

הוסיפו דפים לתרשים לפי הטבלה. **אל תבצעו שמירה!** להוספת דף, לחצו לחיצה ימנית על שטח ריק בתרשים ובחרו New ASP Page. תנו שמות לדפים לפי הכתוב בטבלה. לאחר הוספת הדפים, הזיזו את הדפים על פני שטח התרשים ושימו לב מה קורה לקווי הקישור ביניהם. כאשר אתם מקרבים דף לדף אחר, הם מתחברים בקו. לקווים אלה **אין משמעות מעשית** מעבר לתרשים. למרות זאת נציב את הדפים בצורה שתבטא את ההיררכיה, או את מסלול המעבר של הגולש ביניהם, כמודגם בתרשים הבא.

מהדף הראשי (Home.html) יכול הגולש לעבור לחלון חיפוש ספר (searchbook.asp), או לראות את מצב סל הקניות שלו (showbasket.asp). מדף החיפוש (searchbook.asp) מגיע הגולש לדף התוצאות (booklist.asp, רואים זאת בתרשים), ומדף התוצאות, שהוא רשימה תמציתית, יכול הגולש לפתוח דף פרטים מלא על כל ספר מהרשימה (bookdetail.asp).

צרו תרשים אתר הדומה לתרשים הבא:



לאחר שסיימתם לתכנן את תרשים האתר, לחצו על סמל השמירה בסרגל הכלים (ציור של דיסקט), ושימו לב מה קורה: Visual InterDev, יוצר למענכם דפי ASP לפי התרשים, ומוסיף אותם לפרויקט. זה כמובן מאוד נוח ושימושי. זו אחת הדרכים להוסיף דפים לאתר, אך כמובן לא היחידה.

תרשים האתר הוא לא רק כלי מעולה לראייה כוללת של האתר ולהוספת דפים, אלא גם כלי עזר לכתיבה ותחזוקה. רוצים לערוך את home.html? לחצו לחיצה כפולה על הסימול שלו בתרשים, והוא ייפתח במצב עריכה. ניתן היה גם לחפש אותו בחלון Project Explorer ולהביאו משם, אבל להביא דפים מתוך התרשים זה מאוד נוח ויעיל.

לחצו לחיצה כפולה על סימול דף הפתיחה בתרשים home.html. נפתח חלון העריכה ושימו לב לשלוש הכרטיסיות בחלק התחתון. נחזור על תפקידיהם בקצרה:

Source - מראה לכם את הקובץ כמו שהוא, טקסט. זה המקום לכתוב קוד.

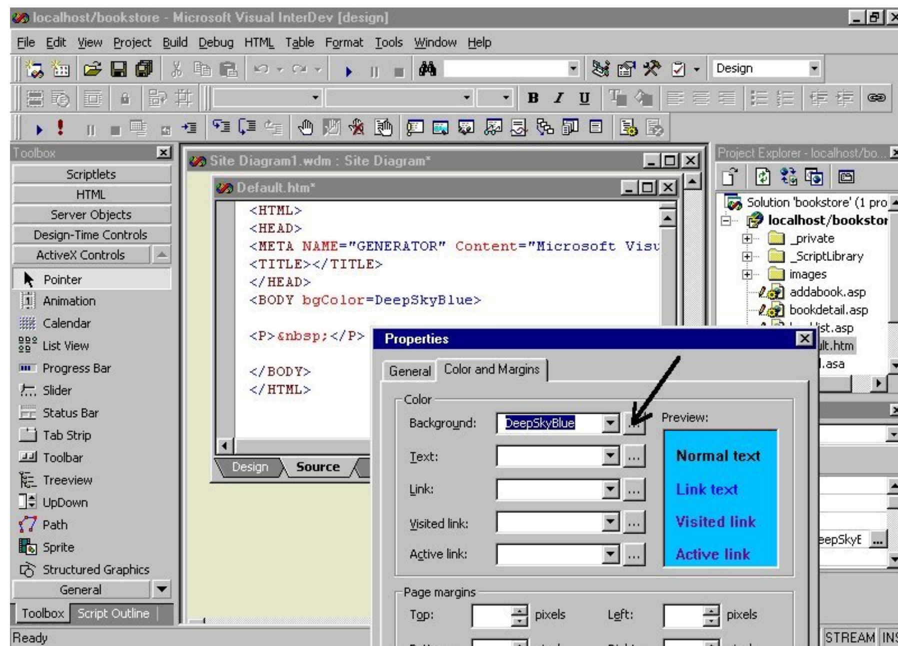
Design - נותן מראה כללי של הדף, וגם מרשה לעשות שינויים גרפיים.

Quick View - מראה את הדף כפי שהוא ייראה לגולש בדפדפן, כולל תפקוד של תוכניות צד-לקוח.

בחרו במצב Design. אנו רואים קוד HTML פשוט ומוכר. למי שהקוד לא מוכר, מומלץ לקרוא ספר בנושא HTML ולהכין לפחות אתר אינטרנט סטטי אחד לפני הצלילה לנושא

ASP. במקום פנוי כלשהו בחלון Design, לחצו לחיצה ימנית בעכבר ובחרו Properties. יופיע חלון תכונות הדף, ובו פרטים רבים שנדון בהם בהמשך.

בחרו בכרטיסייה Colors and Margins, ובחרו צבע רקע לדף הפתיחה.



לחיצה על לחצן שלוש הנקודות המסומן בחץ מביאה חלון בחירת צבעים. מאוד נוח.

לאחר הקשת אישור, בדקו שאכן משהו השתנה: לתג <body> נוספה הגדרת צבע. עברו למצב Design, ותראו שצבע החלון השתנה לצבע שבחרתם.

הישארו במצב Design. שימו את העכבר בראש הדף והקלידו בעברית כותרת לדף הפתיחה: "ספרוני - חנות הספרים שלי באינטרנט". לאחר שכתבתם, סמנו את כל הטקסט והשתמשו בסרגל הכלים כדי למרכז, להגדיל ולצבוע אותו. ניתן כמובן לבחור גופן (font) והדגשות (שמן, נטוי, קו תחתון), ממש כמו ב- Word.

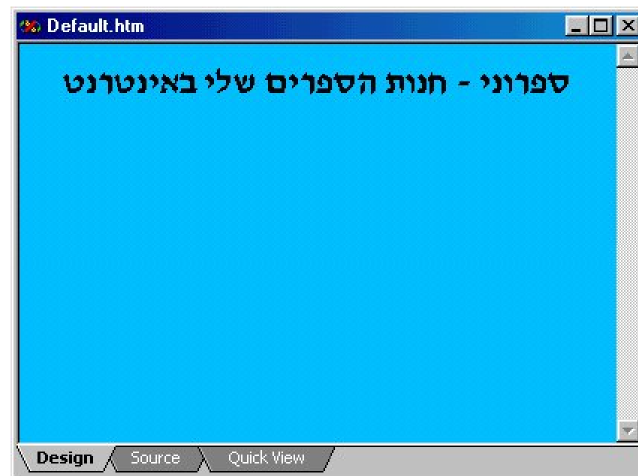


שאלה למחשבה: מדוע גדלי האותיות הניתנים לבחירה הם רק מ-1 עד 7 ולא כמו ב- Word?

תשובה: מפני ש- Visual InterDev מתרגמת את פקודות העריכה שלנו ל-HTML, וגדלי אותיות ב-HTML, נעים בין 1 ל-7.

לאחר העריכה, עברו למצב Source וראו את שינויי הקוד שבוצעו אוטומטית. נכון, שום דבר מדהים. יכולנו לעשות זאת לבד.

במצב Design תוצאת עבודתכם צריכה להיראות דומה לזו:



שמרו את עבודתכם וסגרו את החלון. חזרתם אל תרשים האתר. חזרו על הפעולות שביצעתם בדף הראשי, עבור כל אחד מהדפים: קביעת צבע רקע, וכתיבת כותרת בעברית.

כתבו: "דף חיפוש ספרים", "דף הוספת ספר", "רשימת ספרים" וכן הלאה.

יצירת קישור בין דפים

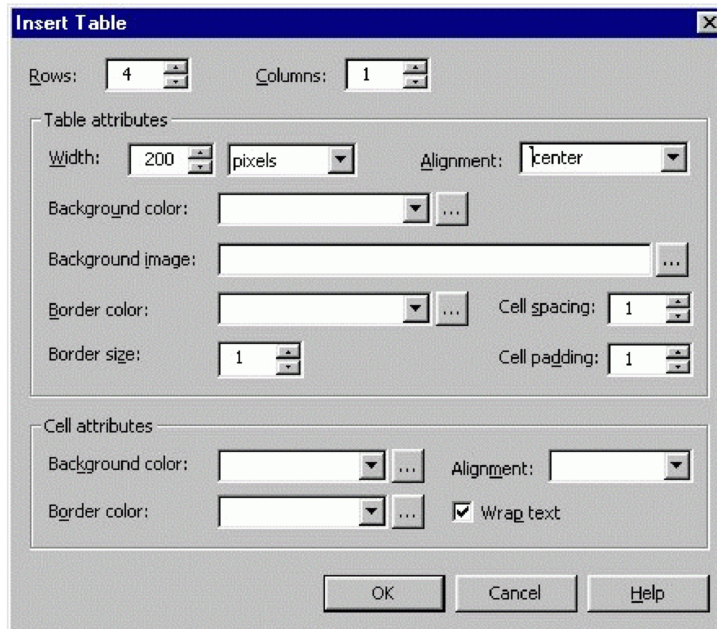
בסעיף זה נלמד לבצע קישורים בין דפי האתר. כלומר: מעבר מדף אחד לדף שני. נכון, זהו בדיוק אותו קישור ישן וטוב שאנו מכירים. רק דרך הביצוע שלו מעט קלה יותר, ופחות חשופה לטעויות. ניגש לעבודה.

מתוך תרשים האתר (נקודת מוצא נוחה תמיד) פתחו את דף הבית, במצב Source.

הציבו את הסמן מעט לפני התג </body>.

בתפריט Table בחרו Insert Table. תופיע תיבת דו-שיח Insert Table. הקלידו 4 שורות, עמודה אחת, רוחב 200 פיקסל, ממורכז.

ראו בתרשים הבא :



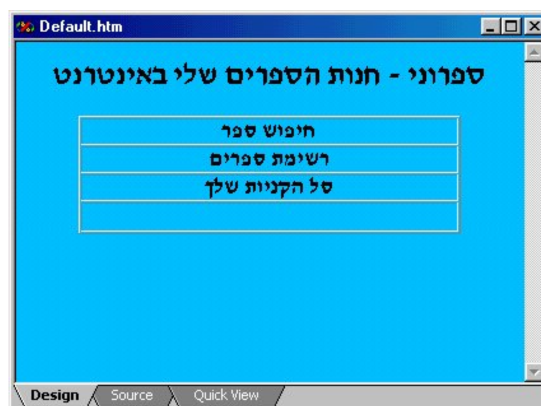
לחצו אישור, וודאו, במצב Source, שבדף הפתיחה שלכם אכן נוצרה טבלה.

עברו למצב Design. רואים את הטבלה שיצרתם. הציבו את הסמן בשורת הטבלה הראשונה, וכתבו: " חיפוש ספר".

בשורה שנייה כתבו : "רשימת ספרים".

בשורה שלישית כתבו : "סל הקניות שלך".

מרכזו את השורות, כך שהתוצאה תיראה כך :

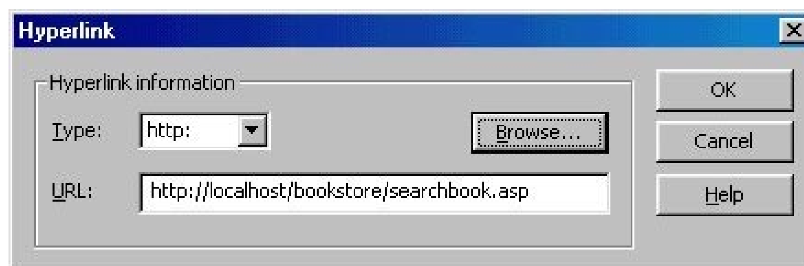


כעת נקשר כל אחת מהשורות לאחד מהדפים שבנינו. זה הולך להיות קל מאוד.

הישארו במצב Design, סמנו בעזרת העכבר את השורה "חיפוש ספר", וכשהיא מסומנת (selected), לחצו על סמל הקישור.

זהו הסמל הימני ביותר בסרגל הכלים (ציור של שרשרת). אם הוא אפור ולא פעיל, סימן שלא סימנתם את הטקסט כמו שצריך.

אם הכל בסדר, תופיע תיבת דו-שיח Hyperlink ובה אתם מתבקשים לכתוב או להצביע להיכן יבוצע הקישור.



לחצו על הלחצן Browse. תופיע תיבת דו-שיח נוספת. פתחו את הפרויקט שלכם, (לחיצה על +) ותראו את כל דפי הפרויקט הקיימים. בחרו בדף searchbook.asp, ואשרו את שני החלונות. נוצר קישור למילים "חיפוש ספר" ואנו יודעים זאת לפי הקו התחתון מתחת לטקסט הנבחר.



הערה:

כך ניתן ליצור קישור גם לאתר אחר לגמרי!

שמרו את הדף והריצו: לחצו לחיצה ימנית במקום בדף ובחרו View in Browser.

ייפתח Explorer עם הדף שלכם. לחצו על הקישור "חיפוש ספר", ודף חיפוש הספרים יופיע.

לחצו back בדפדפן וחזרו לדף הבית. עובד? יופי.

בדקו את קוד HTML שנוצר. ודאו שאתם מבינים אותו.

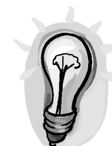
שימו לב! במקום להריץ את הדף באמת, ניתן לעבור ל- Quick View, ולראות בדיוק את אותו דבר.

כעת, באותה דרך, בצעו בעצמכם קישור לשני הדפים הנותרים. התוצאה צריכה להיראות כך:



פינת הגאון - קישור לדף קיים

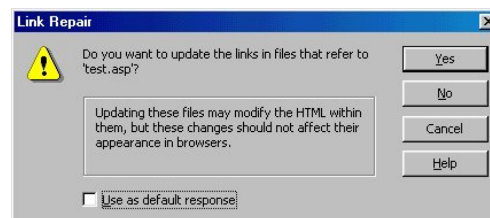
1. מסמנים טקסט או אובייקט אחר.
2. לוחצים על סמל הקישור (ימני ביותר, שרשרת), ובוחרים לאן לקשר.



טיפ:

כיצד משנים שם של דף קיים ב- Visual InterDev ?

אם תבחרו דף ב- Project Explorer ותלחצו לחיצה ימנית, תראו שיש אפשרות Rename, אלא שהיא אפורה וחסומה. זה בגלל הקשר שמקיים InterDev בין העותק המקומי לעותק בשרת. כדי לאפשר שינוי שם, יש לנתק זמנית את הקשר הזה. לחצו לחיצה ימנית על שם הקובץ, ובחרו Release working copy. אחר כך בחרו באפשרות Rename ושנו את שם הקובץ כרצונכם. לאחר שתאשרו, יופיע חלון דומה לזה:



האם לעדכן את שינוי השם לכל המקומות בהם הוזכר דף זה? בדרך כלל תהיה התשובה כן. זה שירות יפה שיכול לחסוך הרבה מאוד עבודה, והרבה באגים.

פרק 4

קליטת מידע מהמשתמש

שיטת GET

כדי לקלוט מידע מהמשתמש, יש צורך ליצור "טופס". טופס (form) נוצר על ידי אוסף תגי HTML המיועדים בדיוק לכך: לתחום מספר שדות קלט ביחד, ולאפשר את שליחתם לשרת.

נבנה יחד דוגמה לכך. צרו דף חדש וקראו לו GET1.ASP. ליצירת דף חדש לחצו לחיצה ימנית על שם הפרויקט, ובחרו Add < Active Server Page. לתוך הדף שקיבלתם הקלידו את השורות הבאות:

```
<%@ Language=VBScript %>
<html>
<head>
<meta NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</head>
<body>
<STRONG>Please Enter Your Name:</STRONG>
<form action="get2.asp" method="get" id="form1" name="form1">
First Name: <INPUT type="text" name=fname><BR>
Last Name: <INPUT type="text" name=lname><BR>
<INPUT type="reset" value="Reset" id=reset1 name=reset1>
<INPUT type="submit" value="Submit" id=submit1 name=submit1>

</form>
</body>

</html>
```

השורה החשובה ביותר בקטע זה היא השורה המודגשת. בשורה זו אנו מכריזים על הטופס, נותנים לו שם, קובעים לאיזה דף לעבור לאחר ביצוע דף זה, וקובעים את שיטת העברת המידע לשרת. הסבר על השיטות יופיע בהמשך.

לאחר הכרזת הטופס, מופיעים שדות הקלט. בדוגמה שלנו שדות טקסט פשוטים: שם פרטי ושם משפחה. שימו לב לתכונת שדה הקלט name. רק לפי שם זה נוכל לזהות את

השדה בשלב מאוחר יותר! ודאו שאתם כותבים נכון את שם הרכיב, כי תצטרכו לקרוא לו בשם זה מאוחר יותר.

לאחר מכן מופיעים שני לחצנים. האחד מטיפוס reset, שמוחק את כל הנתונים מהטופס. השני מטיפוס submit, ולחיצה עליו תגרום לשליחת כל נתוני הטופס לשרת. השרת מקבל את הנתונים שהוקלדו, וקורא לדף הבא. מיהו הדף הבא? כמובן, זה שביקשנו לקרוא לו בהגדרת הטופס במסגרת התגית <form>: GET2.ASP.

צרו דף חדש, GET2.ASP, והקלידו בו את השורות הבאות:

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY><FONT size=5><STRONG>User Information: </STRONG></FONT><BR>

First Name: <B><%=Request.QueryString("fname")%></b><BR>
Last Name: <B><%=Request.QueryString("lname")%></b><BR>

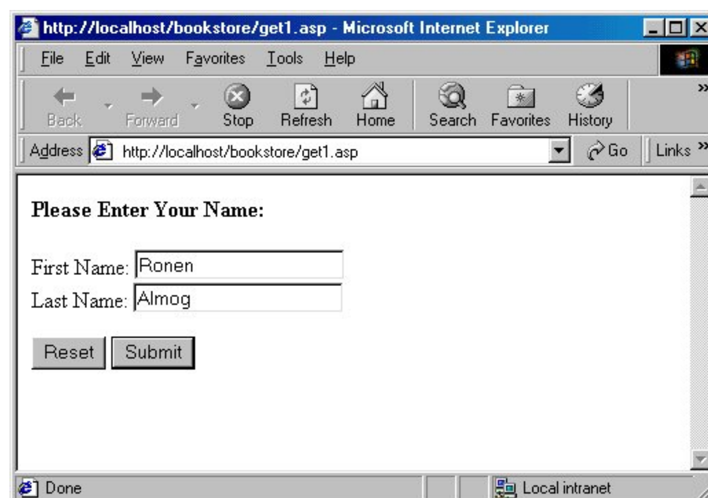
</BODY>
</HTML>
```

הקטעים החשובים מודגשים עבורכם.

לאחר המילים First Name, מופיע קטע קוד ASP, תחום בתוך <% ..... %> כמתחייב:

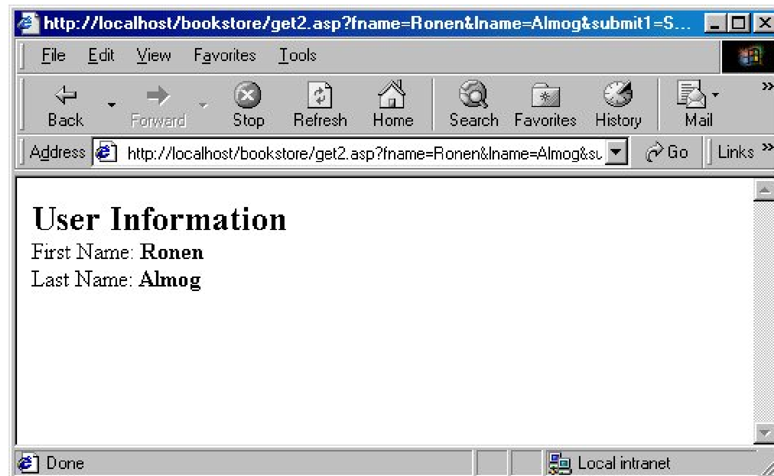
```
<%=Request.QueryString("fname")%>
```

בתוך האובייקט Request, פונים לאוסף QueryString ובתוכו לפריט fname שהוא, למי שכבר הספיק לשכוח, שם שדה הקלט של שדה השם הפרטי בטופס. נריץ את הדף GET1.ASP (לחיצה ימנית על חלון הקוד, ואז View in Browser). נקבל משהו דומה לזה:



הקלידו את שמכם ולחצו על Submit.

זו התוצאה המתקבלת, לאחר לחיצה על הלחצן Submit:



לא עובד?

1. ודאו שלא שכחתם לכתוב את סימן השוויון לפני ה- Request.
2. ודאו ששם רכיב הקלט בדף GET1, זהה לשם אותו אתם קוראים בדף GET2.

לאחר מכן, שימו לב לשורת הכתובת בדפדפן! אתם רואים לא רק את שם וכתובת הדף, כמו בדפי HTML רגילים, אלא גם את מחרוזת ה- http, המחביאה בתוכה את הנתונים ששלחתם. מתוך כתובת זו ניתן ללמוד כיצד השיטה GET עובדת: לאחר כתובת הדף, היא משרשרת את כל הפרמטרים שיש להעביר לדף הבא, עם סימן & ביניהם. כמו בדוגמה שלנו:

http://localhost/bookstore/get2.asp?fname=Ronen&lname=Almog&submit1=Submit

שיטה זו נקראת URL Encode, הצפנת הפרמטרים בתוך שורת הכתובת. הפקודה Request.QueryString מחלצת מתוך שורה זו ערך מסוים. אם נכתוב

fname=Request.QueryString("fname")

תחפש הפונקציה בתוך מחרוזת הכתובת את המילה fname, ותחזיר לנו את הערך שבא מייד אחריה. במקרה שלנו: Ronen.

זו שיטה לא רעה, וכפי שראינו היא עובדת היטב. אלא שיש לה כמה מגרעות:

1. המשתמש רואה את המחרוזת המופיעה למעלה, ולא תמיד זה רצוי.
2. לא רק המשתמש, אלא גם כל התחנות בהן עובר המידע בדרך לשרת, "רואות" את המידע כמחרוזת פשוטה. לא בריא להעביר כך מספרי כרטיסי אשראי, למשל.

3. מחרוזת http מוגבלת באורכה ל- 1000 תווים בערך. מכאן שאם יש לנו בטופס שדות רבים ו/או טקסטים ארוכים, אנחנו בצרות. ברור גם שלא נקבל כל הודעת שגיאה, אלא פשוט הטקסט יקוצץ בנקודה מסוימת.

לכן כדאי לדעת גם על השיטה השנייה לשליחת מידע לשרת, שיטת post.

שיטת Post

גם דוגמה זו תהיה מורכבת משני דפי ASP. האחד להזנת נתונים, והשני לעיבוד המידע המתקבל. הוסיפו לפרויקט שלנו דף ASP חדש: post1.asp.

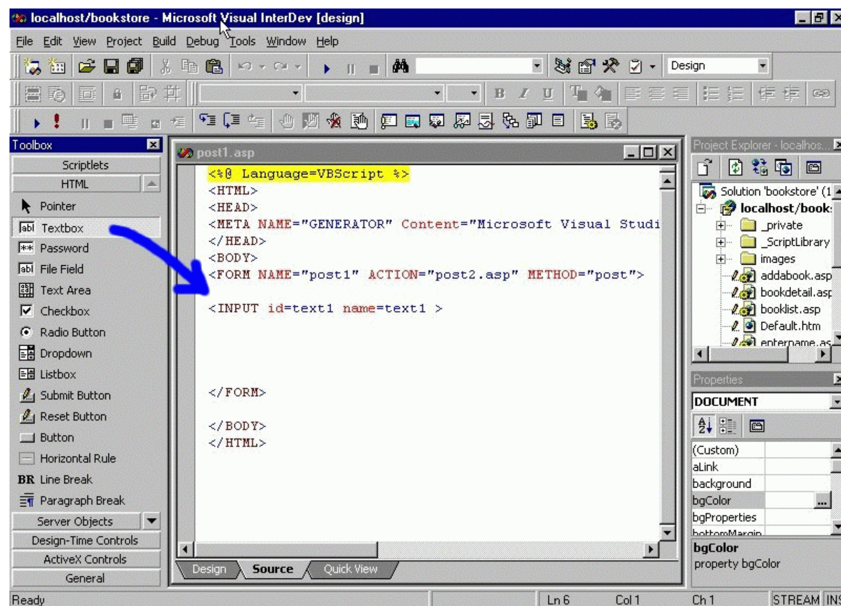
בדף זה, לאחר התג <body> הוסיפו הגדרת טופס כזו:

```
<FORM NAME="post1" ACTION="post2.asp" METHOD="post">
```

הקישו Enter מספר פעמים כדי להכין מקום לתוכן הטופס, וסיימו אותו ב- </Form>.

ננצל את ההזדמנות ונכיר עוד כמה מאפשרויות הקלט הקיימות. ודאו שחלון Toolbox פתוח (בדרך כלל בצד שמאל של המסך). אם לא, פתחו אותו בעזרת התפריט: View > Toolbox. כעת נגרור מארגז הכלים רכיב Textbox אל תוך הטופס שלנו. הקפידו להניח אותו בתוך הטופס ולא מחוצה לו. כלומר, בין התג <Form> לתג </Form>. בדיקה נוספת: עוברים ל-design ורואים אם החיצים הצהובים מקיפים את האובייקט שלנו.

כך גוררים אובייקט מארגז הכלים לחלון העריכה:

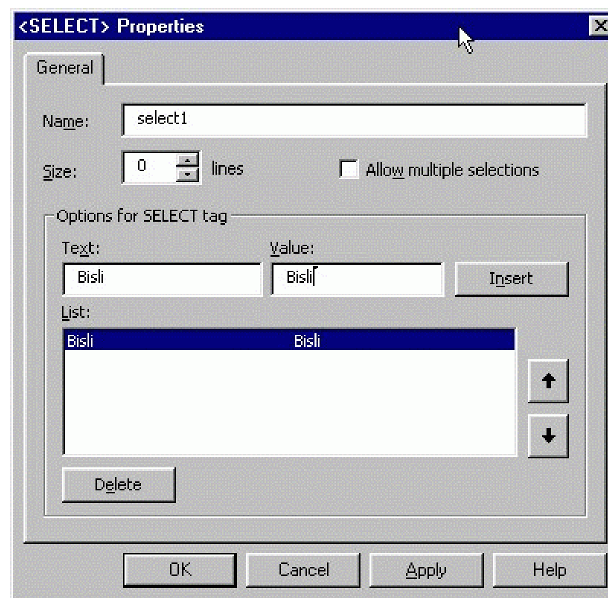


הציבו את הסמן לפני תיבת הטקסט וכתבו משהו, כגון: Please enter your name.

אחרי השורה הוסיפו
 כדי לעבור לשורה הבאה. לעצלנים במיוחד: שימו לב שגם את תג זה ניתן לגרור מארגז הכלים. הוא נקרא Line Break והוא נמצא בחלק התחתון. כעת קחו רכיב Text Area והניחו אותו בעדינות בהמשך הטופס. עברו למצב Design והגדילו אותו למלוא רוחב החלון. הוסיפו לו גם כותרת כלשהי כדוגמת:

Please Tell me your life story

כעת, הוסיפו גם רכיב Dropdown. רכיב זה נקרא לעיתים "תיבת קומבו" (Combo Box), ובקיצור "קומבו". הוא שימושי ביותר, והוא נותן למשתמש לבחור בקלות בין כמה אפשרויות, בלי לבזבז שטח מסך יקר. לאחר שהוספתם אותו, עברו למצב Design, לחצו לחיצה ימנית על הקומבו החדש, ובחרו Properties. יופיע חלון כזה:



הקלידו "Bisli" בשני השדות text ו- value, ולחצו על הלחצן Insert.

חזרו על הפעולה עבור מוצרים אחרים, כך שיהיו לכם ברשימה כמה מוצרים. לסיום, לחצו OK וחזרו למצב Source כדי לראות מה נכתב.

שפצו מעט את מראה הקוד, שייראה בערך כך:

```
<SELECT id=select1 name=select1 style="HEIGHT: 22px; WIDTH: 200px">
<OPTION selected value=0>Please select a product</OPTION>
<OPTION value=Bisli>Bisli</OPTION>
<OPTION  value=Bamba>Bamba</OPTION>
<OPTION value=Shoko>Shoko</OPTION>
<OPTION value=Krembo>Krembo</OPTION>
</SELECT>
```

שימו לב לשורת option הראשונה שהוספתי ידנית. בגלל המילה selected שורה זו תופיע בקומבו כברירת מחדל.

הרכיב "קומבו" משמש לבחירת ערך אחד מתוך כמה. כך הוא נראה במצב סגור:

Please select a product

וכך הוא נראה, כאשר לוחצים על סמל החץ ופותחים את הרשימה. ניתן לטייל על פני הרשימה ולבחור ערך. הקשת האות הראשונה מביאה אותנו לערך הראשון המתחיל באות זו.

Bamba
Please select a product
Bisli
Bamba
Shoko
Krembo

האמת היא שהרכיב מסתיר מעט יותר מזה. המשתמש רואה רשימה, אך בעצם יש כאן שתי רשימות מקבילות. האחת לעיני המשתמש, והשנייה, לשימושנו, המתכנתים. כאשר הגולש מסמן "במבה", מה אתם רוצים לקבל? בדרך כלל תרצו לקבל את הקוד של המוצר, ולא את שמו. הקוד הוא זה שייכנס להזמנה ויישמר במסד הנתונים, ממש כמו שהקופאית בסופרמרקט אינה מקלידה "לחמניה עגולה עם שומשום למעלה" אלא רק את הקוד שלה.

מכאן קל יותר להבין את המונחים Text ו- Value. הטקסט הוא הערך המוצג לגולשים. Value הוא הערך האמיתי, המסתתר מאחורי הטקסט, אותו אנחנו נקבל כשנשאל, בדף הבא, מה בחר הגולש. בדוגמה שלנו, למען הפשטות, ויתרנו על השימוש בקוד, והקלדנו ערך מילולי בשני המקומות. התוצאה: אנחנו נעביר לדף הבא את שמו של המוצר. בדרך כלל, השתדלו להעביר רק קודים.

לסיום, גררו מארגז הכלים את הלחצנים reset ו-submit.

קוד הדף כולו, post1.asp:

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>
<FORM NAME="post1" ACTION="post2.asp" METHOD="post">

Please Enter your name:<BR>
<INPUT id=text1 name=text1 ><BR>
```

```
Please tell me the story of your life:<BR><TEXTAREA id=textarea1 name=textarea1
style="HEIGHT: 50px; WIDTH: 435px"> </TEXTAREA>
<BR><BR>

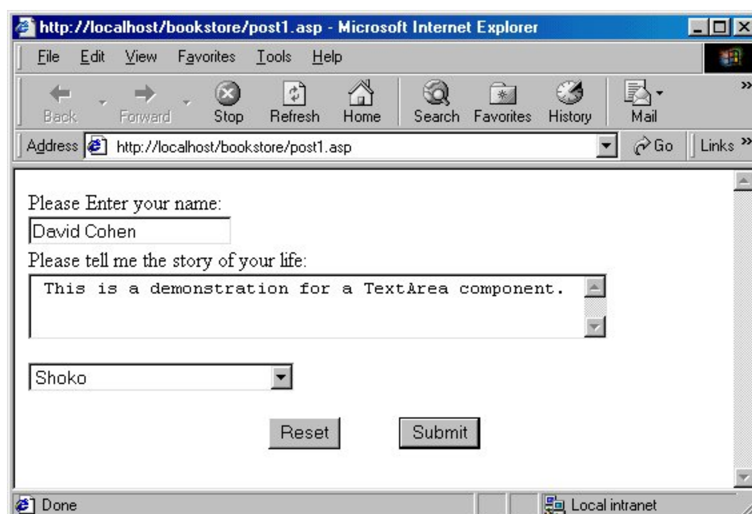
<SELECT id=select1 name=select1 style="HEIGHT: 22px; WIDTH: 200px">
<OPTION selected value=0>Please select a product</OPTION>
<OPTION value=Bisli>Bisli</OPTION>
<OPTION  value=Bamba>Bamba</OPTION>
<OPTION value=Shoko>Shoko</OPTION>
<OPTION value=Krembo>Krembo</OPTION>
</SELECT>
<BR><BR>

<P align=center>
<INPUT type="reset" value="Reset" id=reset1 name=reset1>
<INPUT type="submit" value="Submit" id=submit1 name=submit1>
</P>

</FORM>
</BODY>

</HTML>
```

כשתריצו את הקוד, החלון שלכם צריך להיראות דומה לזה:



להזכירכם, דוגמה זו נועדה להדגים העברת נתונים לחלון הבא בשיטת post. לכן נבנה כעת דף ASP חדש שייקרא post2.asp. דף זה ייקרא על ידי post1 ובו נקלוט את הערכים שהמשתמש הזין.

הוסיפו לפרויקט דף ASP חדש בשם post2.asp, והקלידו לתוכו את הקוד הבא :

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

Thank You, <b> <%=Request.Form("text1")%> </b>, for your order.<BR>
<BR>
The story of your life is this:<BR>
<b> <%=Request.Form("textarea1")%> </b><BR><BR>
The product you selected is <b> <%=Request.Form("select1")%>
</b><BR><BR>

Thank you for your support, and come again.<BR>

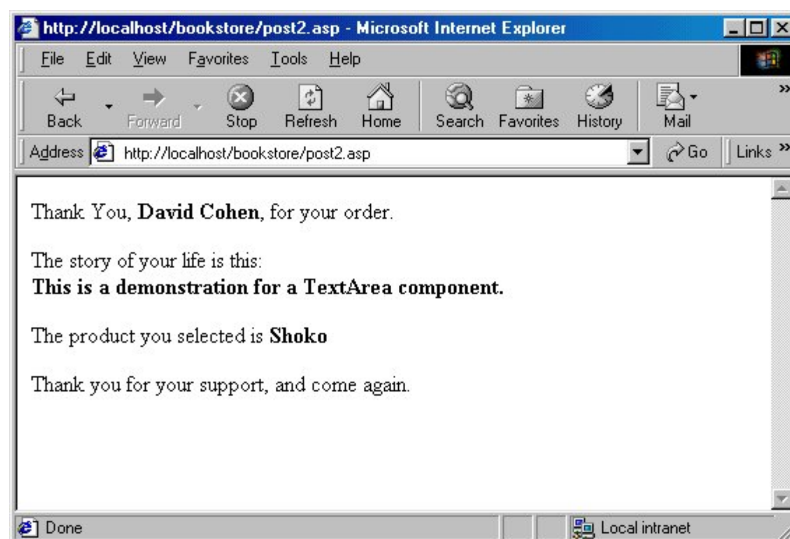
</BODY>
</HTML>
```

הטקסט המודגש הוא זה שבשבילו טרחנו להגיע עד הנה. זו השיטה לקבל מידע מדף קודם בשיטת post.

```
<%=Request.Form("textarea1")%>
```

שימו לב! לא לשכוח את השווה הקטן לפני המילה Request. בלעדיו - לא עובד.

הריצו את post1.asp, מלאו פרטים, לחצו submit וראו את התוצאות בדף post2.asp. אם יש הודעות שגיאה - התמודדו איתן. נסו להבין היכן הטעות. כל קטעי הקוד בספר זה נבדקו ונמצאו "דוברי אמת".



אם הכל עובד יפה, שימו לב לשורת הכתובת ב- post2.asp. שורת הפרמטרים שהיתה בדוגמה הקודמת - נעלמה. התגברנו על מגרעה של חשיפת הפרמטרים לעיני כל. התגברנו גם על מגבלת 1000 התווים, אבל לא ניתן לראות זאת.

סיכום

ראינו שתי שיטות להעברת פרמטרים בין דף אחד למשנהו. שיטת GET ושיטת POST. הראינו ששיטת POST טובה יותר מכמה סיבות, ואינה קשה יותר להפעלה. לכן, בדרך כלל נשתמש בה.

פינת הגאון - קבלת נתונים מטופס

שיטת GET:

בטופס השולח נתונים:

```
<FORM NAME="get1" ACTION="get2.asp" METHOD="GET">
```

קליטת ערכים

```
</FORM>
```

קבלת הנתונים בטופס המקבל:

```
var=Request.QueryString("fname")
```

שיטת POST:

בטופס השולח נתונים:

```
<FORM NAME="post1" ACTION="post2.asp" METHOD="POST">
```

קליטת ערכים

```
</FORM>
```

קבלת הנתונים בטופס המקבל:

```
var =Request.Form("fname")
```

פרק 5

התקשרות למסדי נתונים

עד כה שיחקנו מעט, העברנו נתונים מטופס (שהוגדר במסגרת דף ASP) לדף ASP, והיה נחמד. אך צריך להיות ברור, בשלב זה, שעיקר כוחו של יישום אינטרנט (Internet Application), הוא לעבוד מול מסדי נתונים. מסד הנתונים יכול להכיל את פרטי המוצרים שאנו מוכרים, את פרטי המבקרים באתר, את המידע אותו אנו מפרסמים וכל דבר אחר, שאותו אין טעם או אפשרות לכתוב בדף HTML רגיל.

באיזה מסד נתונים לבחור?

בספר זה נעבוד עם מסד הנתונים Access 97, מפני שהוא זמין, זול ונוח מאוד לעבודה. באתרים בהם יש עומס רב של מבקרים בו-זמנית, או שכמות המידע בהם היא רבה, נבחר מסד נתונים חזק יותר, כדוגמת SQL Server, שמסוגל לעמוד בעומסים גבוהים. אם נבחר ב- SQL Server, נהנה גם מיתרונות, כמו: יכולת גידול (Scalability), תמיכה בריבוי מעבדים, פתרון מקצועי לגיבוי, שרידות גבוהה והתאוששות מנפילות. נסו לחשוב על כך: כיצד בדיוק מגבים מסד נתונים של אתר מכירות מקוון? אנשים מבצעים קניות 24 שעות ביממה (ביום ובלילה). אסור להוריד את השרת. מה ייכנס לגיבוי, ומה לא? האם ייתכן שהזמנת הלקוח תגובה, והתשלום לא? לכל שאלות אלה ולעוד רבות, נותן SQL Server פתרונות מסודרים. Access, מה לעשות, לא.

עם זאת, חשוב לדעת שהמרת מסד נתונים של Access ל- SQL Server היא דבר קל ופשוט, לכן ניתן בהחלט להתחיל את הפיתוח ב-Access, ולגדול אחר כך.

שפת SQL

מול מסד הנתונים שלנו "נדבר" בשפת SQL. זו שפה סטנדרטית לעבודה מול מסדי נתונים, ויש לה שני יתרונות גדולים:

1. פשטות. SQL היא שפה קלה מאוד, דומה לאנגלית.
2. תאימות. כל מסדי הנתונים המובילים כיום עובדים ב-SQL. מכאן שהחלפת מסד נתונים לא צריכה לדרוש כתיבת היישום מחדש.

לשפת SQL יש גרסאות השונות מעט זו מזו. הבסיס לכולן היא השפה התקנית, הנקראת ANSI SQL-92. אלא שמאז הכרזתה של גירסה מחייבת זו, המשיכו החברות לפתח ולשכלל את מוצריהן, ורובן מציעות היום גרסאות משופרות של השפה. גרסת אורקל, לדוגמה, נקראת PL-SQL, ואילו גרסת מיקרוסופט הפועלת ב- SQL Server נקראת Transact SQL או בקיצור T-SQL. שתי גרסאות אלה הן הרחבה (Superset) של ANSI SQL-92. כלומר, כוללות אותה ויותר מכך.

במה כדאי להשתמש?

תלוי. אם הקמתם חברת מרפסת שעתידה לא ברור, בניתם מסד נתונים ב-Access וצפוי לכם גידול משמעותי בנפח הנתונים – כדאי לכם להיצמד לתקן, כי מי יודע לאיזה מסד נתונים תעברו, וכדאי להיות תואמים לכולם. אם לעומת זאת השקעתם עשרות אלפי דולרים במסד נתונים מסחרי, ואתם כבר "מחותנים" איתו – כדאי לפחות ליהנות מהשיפורים שהוא מציע.

איך ללמוד SQL? מספר כמובן. אני ממליץ על ספרו של רז הייפרמן "**בסיסי נתונים טבלאיים ושפת SQL**" בהוצאת הוד-עמי. לא רק SQL תמצאו שם, אלא גם ניתוח מערכות, איך לתרגם את צרכי הלקוח למבנה טבלאי, איך ליצור מסד נתונים יעיל וקומפקטי, איך לשפר ביצועים בקלות, ועוד ועוד.

עזרה מיידית בכל שאלת SQL דחופה שיש לכם, תקבלו ממקור בלתי צפוי: קובץ העזרה של Access. בקובץ זה, שהוא אחד הטובים בתעשיה כולה, תמצאו תיעוד מפורט של כל פקודות SQL הקיימות, הסברים באנגלית קלה, והעיקר דוגמאות. ודוגמה, כידוע, טובה מאלף מילים.

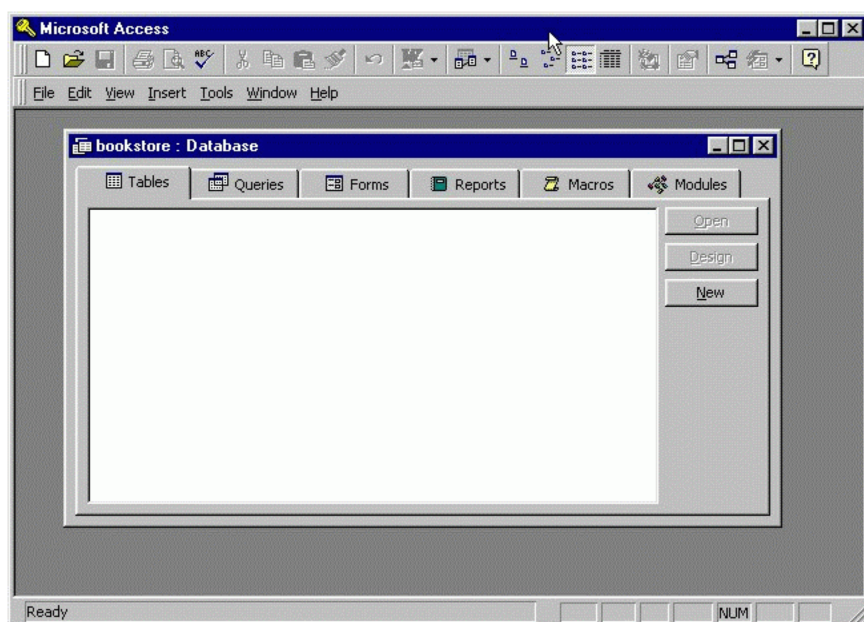


הערה:

כל ההוראות וההסברים בספר זה מתייחסים ל- Access 97, אך נכונים לחלוטין גם ל- Access 2000.

בניית מסד הנתונים

הפעילו את Access וצרו מסד נתונים חדש, שייקרא bookstore.mdb. שמרו אותו במדור c:\bookstore, יחד עם כל קבצי הפרויקט שלנו. נקבל חלון שנראה כך:

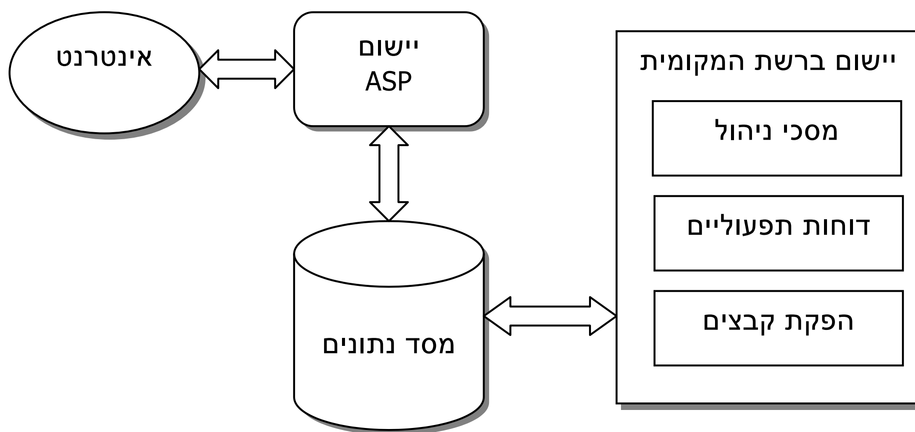


ב- Access 2000 החלון נראה מעט אחרת, אבל ההבדלים קוסמטיים בלבד. ניתן לבצע את הפרויקט גם ב- Access 2000 ללא שום בעיות.

כפי שניתן לראות, שפת הממשק שבחרתי היא אנגלית. אם במחשב שלכם התפריט מכיל את המילים "קובץ, עריכה, צפייה...", זה סימן שאתם עובדים בממשק עברי. כדאי לעבור לממשק אנגלי, אך לא חובה. בחרו בתפריט **כלים** > **אפשרויות**. מופיע חלון עם מספר כרטיסיות. בחרו בכרטיסיה "עברית" והחליפו את שפת הממשק לאנגלית. זה כמובן לא אומר שום דבר על **הנתונים**, שיאוחסנו במסד הנתונים. אלה יכולים כמובן להיות בעברית או אנגלית, ללא שום הבדל.

מילה על Microsoft Access. זהו כלי מופלא. קל לשימוש במידה הגורמת להתמכרות. בדרך כלל יש הפרדה ברורה בין כלי פיתוח, כגון: Visual Basic, Delphi, PowerBuilder, לבין מסדי נתונים כגון: Informix, Oracle, SQL Server ודומיהם. Access שונה בכך שהוא הכל ביחד. הוא גם מסד נתונים, גם כלי מעולה לשאילתות וגם כלי פיתוח קדמי, שניתן לפתח באמצעותו יישום PC רגיל, עם תוכניות, תפריטים, סרגלי כלים, חלונות וכו'. ריבוי תפקידים זה, משתקף בחלון מסד הנתונים החדש שיצרנו. החלון מחולק לשש כרטיסיות. אנחנו, כמפתחי אינטרנט, משתמשים רק בשתי הכרטיסיות השמאליות: טבלאות (Tables) ושאילתות (Queries). יתר הכרטיסיות משמשות לפיתוח תוכניות באמצעות Access, שאינן מתאימות לאינטרנט.

עם זאת, ייתכן שתעשו שימוש גם בחלקים אלה. מצד אחד ניגש כל העולם למסד הנתונים שלכם דרך האינטרנט, אך מצד שני מישהו צריך לנהל את המלאי? מישהו צריך להכין הזמנות לאספקה? מישהו צריך להדפיס רשימת מוצרים להזמנה מהספקים? מישהו צריך להכין קבצי חיוב לחברות האשראי? אני מקווה שהמישהו הזה הוא לא אתם, אך בכל מקרה, את כל המשימות אלו, קל יותר לעשות ביישום Access רגיל, ולא דרך האינטרנט. זה מוביל אותנו למבנה הבא:

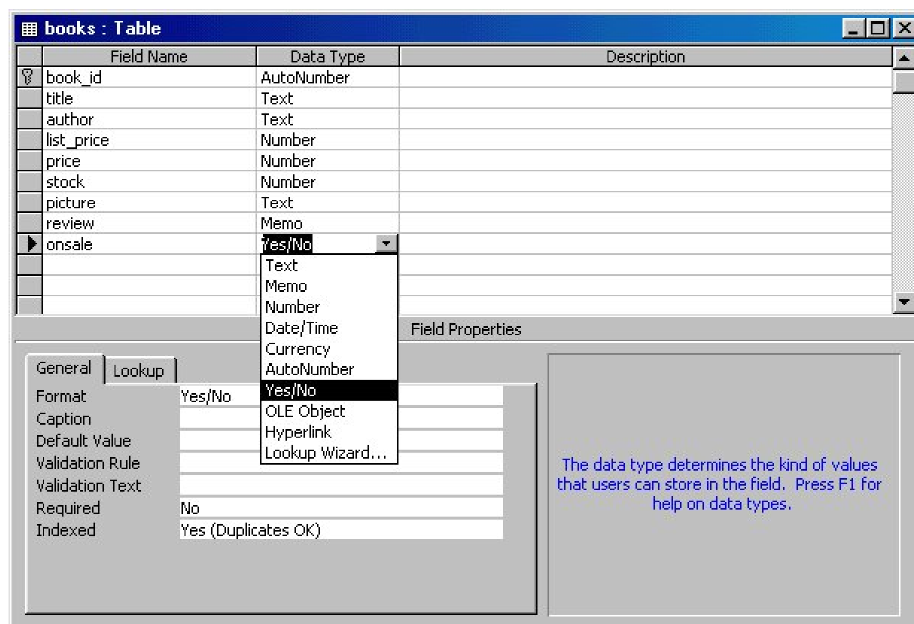


מסד הנתונים משותף, ו"נראה" על ידי שני הצדדים: צד האינטרנט, שהוא כמובן יישום ה-ASP שלנו, ומאחוריו רשת האינטרנט כולה. בצד היישום נמצאת תוכנית PC (זו יכולה להיות תוכנית הכתובה ב-Visual Basic), העובדת ברשת מקומית או במחשב בודד, ואחראית על פעולות שנוח יותר לבצע בתוכנית רגילה.

צד האינטרנט, כאמור, משתמש רק בטבלאות ובשאילתות. את יישום ה-PC תקימו, אתם או אחרים, בעזרת יתר הכרטיסיות.

כעת לעבודה. ניצור טבלה ראשונה: טבלת ספרים.

סמנו את הכרטיסיה Tables ולחצו על לחצן New. כך יוצרים טבלה חדשה. מופיע חלון אפשרויות, בחרו Design View. מופיע חלון תכנון הטבלה. בעמודת Field Name הקלידו book_id (זה יהיה קוד הספר). בעמודת Data Type בחרו ב- AutoNumber (זה מספור אוטומטי). עבור כל ספר חדש שנוסיף, יווצר מספר חדש, גדול ב-1 מקודמו (זה חוסך טעויות ועושה סדר). בשדה Description אין צורך לכתוב כלום.



תרשים: בחירת סוג נתונים

השדה הבא: שם הספר. הקלידו בשם השדה title ובסוג הנתונים: Text. שימו לב לחלק התחתון של המסך, Field Properties, שם ניתן לקבוע תכונות נוספות לשדה. למשל, עבור שדה טקסט ניתן לקבוע את אורכו המקסימלי. ברירת המחדל - 50 תווים - תספיק.

השדה הבא: שם המחבר. הקלידו author בשם השדה, ובחרו גם כאן, בסוג Text.

השדות הבאים: מחירים. יהיו לנו שני מחירים: מחיר מחירון, ומחיר "שלנו". הקלידו בשם השדה list_price. שימו לב לקו התחתון המחבר בין המילים. לעולם לא רווחים בתוך שם שדה! סוג הנתונים הוא Number. ב- Field Properties, ניתן לקבוע איזה <u>סוג</u> מספר זה יהיה. בחרו בסוג Single. זהו מספר עשרוני עם מגבלה של בערך 2 מיליארד שקלים. אנחנו חנות זולה, אין לנו בעיה עם מגבלה כזאת.

השדה הבא: price, גם הוא מסוג מספרי, Single.

השדה הבא: כמות במלאי. אנחנו משתדלים לא למכור ספרים שאין לנו, לכן כדאי לשמור על נתון זה. הקלידו בעמודת שם השדה stock ובעמודת סוג הנתונים - Number. הפעם, בחרו בסוג Long Integer, מספר שלם ארוך, כי ספרים יהיו לנו במלאי, יש לקוות, רק ביחידות שלמות.

השדה הבא: תמונת הספר. זה יהיה שדה טקסט, בו נכתוב את שם הקובץ של התמונה. הקלידו בשם השדה picture ובסוג הנתונים Text.

השדה הבא: Review, תקציר הספר. הקלידו בשם השדה review, ובסוג הנתונים הקלידו memo, זהו סוג נתונים המיועד לטקסטים ארוכים. בניגוד לשדה טקסט רגיל, המוגבל ל- 255 תווים, יכול שדה memo להכיל מעל 1GB. שוני נוסף הוא, שבשדה טקסט אנו קובעים אורך מסוים, וכמות זו של בתים נשמרת על גבי הדיסק, בין אם מילאנו אותה במידע ובין אם לאו. memo, לעומת זאת, שומר רק את האורך הנדרש.

השדה האחרון: onsale. שדה מטיפוס בוליאני (כן/לא), המסמן אם זהו ספר במבצע או לא. כברירת מחדל סמנו No.

לפני סגירה, הניחו את הסמן על השדה הראשון, book_id, ולחצו על לחצן המפתח בסרגל הכלים. זה יהפוך את השדה ל"מפתח ראשי". כלומר: שדה ייחודי, בעל ערכים שאינם חוזרים על עצמם, המשמש כ"מספר זהות" לאובייקט המתואר בטבלה.

במקרה שלנו, book_id הוא מספר הזהות של הספר. אמור לי את ה- book_id ואומר לך מהו שם הספר, מה מחירו וכן הלאה. כמו מספר הברקוד בסופרמרקט, או כמו, להבדיל, מספר תעודת זהות.

סגרו את הטבלה. Access ישאל אם לשמור שינויים. ענו "כן" ואז תופיע שאלה: באיזה שם. הקלידו books וזה יהיה שם הטבלה.

ליתר ביטחון, להלן שוב שמות השדות וסוגי הנתונים:

Field Name	Data Type
book_id	AutoNumber
title	Text 50
author	Text 50
list_price	Number Single
price	Number Single
stock	Number Long Integer
picture	Text 50
review	Memo
Onsale	Yes/No (default: No)

ניתן להשאיר את Access פתוח ולעבוד במקביל גם על Visual InterDev. רק דבר אחד אסור לעשות: להשאיר טבלה/ות פתוח/ות במצב Design, כי אז אתם, הלקוחות שלכם, וכל מי שמנסה לפתוח דף ASP וניגש לטבלה פתוחה, יקבל הודעת שגיאה בנוסח:

The table 'books' is locked by the user 'Admin'...

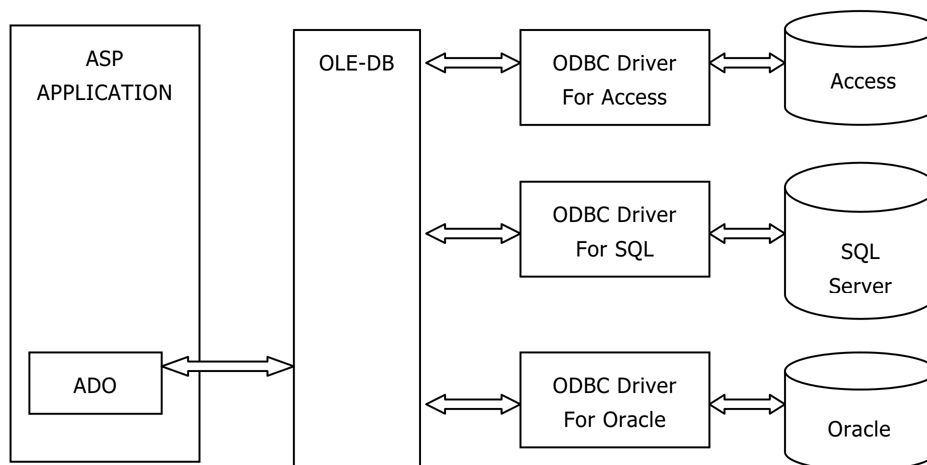
כשהטבלה מוכנה, יש להקליד לתוכה ערכים. סמנו את הטבלה ולחצו על הלחצן Open, או לחצו פעמיים על שם הטבלה. הקלידו מספר שמות של ספרים. לכל אחד הקלידו את שם הספר, שם המחבר, מחיר, מחיר מחירון וכן הלאה.

onsale	review	picture	stock	price	list_price	author	title	book_id
☐	ון והאבסורד.	alice1.jpg	100	45	100	לואיס קרול	עליסה בארץ הפלאות	1
☑	:ל על הדעת.	snark.jpg	1000	12	14	לואיס קרול	ציד הסנארק	2
☑	מוד והמקרא.	shirim_upizmo	10	110	123	ח. נ. ביאליק	שירים ופזמונות לילדים	3
☑	ומד בגן גורים	gan_gurim.jpg	100	8	10	רפאל ספורטה	גן גורים	4
☑	ים בגן שושנה	kippa.jpg	120	129.9	149.9	א. זאב	כיפה אדומה	5
☑	כל גבול וחוק!	maxandmoriz.	1	99.9	120	וילהלם בוש	מקס ומוריץ	6

שימו לב, שאי אפשר להקליד מספר ספר. המספור הוא אוטומטי, כשתקלידו את האות הראשונה בשם הספר, יופיע המספר מעצמו.

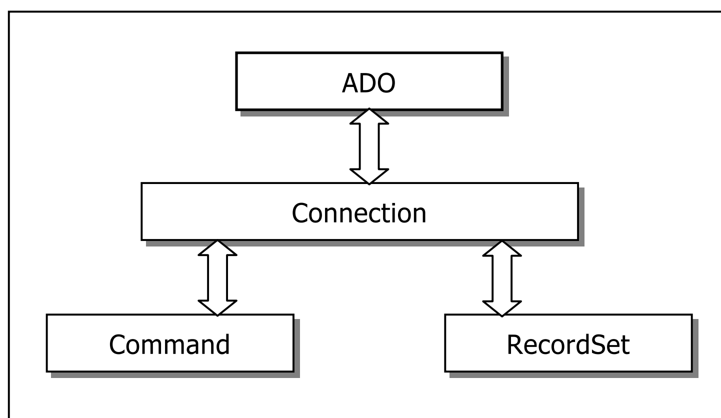
ADO

הקישור למסד הנתונים, נעשה באמצעות האובייקט ADO (ActiveX Data Objects), ומנוע OLE-DB. זהו מודל אובייקטים, המאפשר התחברות למסדי נתונים שונים. באמצעות השימוש בשכבת "תיווך" אנו משיגים שתי מטרות: ראשית, והחשובה ביותר, נחסך מאיתנו העונג המפוקפק לתכנת בשפתו הטבעית של מסד הנתונים. ADO יעשה זאת במקומנו. אנחנו רק נאמר "פתח לי את טבלת העובדים" והוא יתרגם זאת לשפתו של מסד הנתונים. שנית, אם האתר שלנו יהיה הצלחה מסחררת (בטח שיהיה), יבואו אלפי גולשים ונחליט לעבור למסד נתונים אחר, כל מה שנצטרך לעשות הוא להחליף "מתאם" ל-ADO למסד הנתונים החדש. בשפה המקצועית - Driver. כל הפקודות שכתבנו יישארו ללא שינוי, הן מדברות רק עם ADO. התרשים הבא יבהיר זאת:



היתרון במבנה זה ברור - מכיון שהיישום שלנו מדבר רק מול ADO, החלפת מסד נתונים תהיה עניין פשוט יחסית.

נכיר מספר אובייקטים נוספים בעולם ADO: Connection, Recordset, Command.



Connection

קישור למסד נתונים. אובייקט זה יוצר קשר בין תוכנית ASP שלנו, לבין מנהל ההתקן של מסד הנתונים. בין נתוני הקישור יהיו למשל: סוג מסד הנתונים, מיקומו הפיסי, שם משתמש והסיסמה בהם "נכנס" למסד הנתונים ועוד. השם והסיסמה מצריכים הסבר קצר: כל גישה למסד נתונים (בעיקר למקצועיים שבהם, אך גם ל-Access), מצריכה Log In. כלומר, הקלדת שם וסיסמה. כך מבחין מסד הנתונים בין משתמשים שונים, נותן להם זכויות שונות, ומונע אותם מלהפריע זה לזה. Access, כחלק מהקלות הבלתי נסבלת שלו, אינו דורש שם וסיסמה. הוא מכניס אֶתכם אוטומטית כמשתמש Admin, כלומר, מנהל המערכת. רק אם תבקשו במפורש, Access יפעיל את חלון Log In, וידרוש שם וסיסמה.

עבודה כ-Admin נוחה מאוד, כשעובדים בבית או במשרד הפרטי, אך יש לזכור: כל גולש אינטרנט מבצע למעשה קישור למסד הנתונים שלכם, ומתחבר כ- User לכל דבר. לכן, לא יהיה זה נבון לתת לו זכויות של מנהל מערכת. כדאי ליצור עבורו שם משתמש, User במסד הנתונים. משתמש זה יקבל זכויות מאוד מוגבלות, שימנעו ממנו לחולל נזקים. מובן שצריכות להיות לו מספיק הרשאות (זכויות) כדי לבצע את מה שאנחנו רוצים שיעשה. לדוגמה, אם אנחנו עוסקים בחנות ספרים, לא צריכה להיות לגולש זכות למחוק ספר, לא צריכה להיות לו גישה, גם לא לקריאה בלבד, לטבלת הזמנות, ובוודאי לא לטבלאות ספקים ותשלומים.



שאלה: כיצד יכול הגולש למחוק ספר? הרי מעולם לא בנינו, ולא נבנה דף שמאפשר מחיקת ספר. אם כן, מדוע עלינו להגביל את זכויות המשתמש במסד הנתונים?

תשובה: נכון. בדרכים המקובלות, לא יוכל הגולש למחוק ספר, גם לא בטעות. אבל כדי שיבצע פעולות חוקיות, אנו פותחים לגולש דלת אל מסד הנתונים שלנו. בפתח זה יכולים להיכנס גורמים עוינים. הנחת העבודה צריכה להיות תמיד, שהקישור של הגולש נפל בידי האויב. לכן אסור לתת לו לבצע פעולות הרסניות.



שאלה: אם נגביל את זכויות המשתמש, האם לא נפגע בכך בעצמנו? לדוגמה, אסרנו על המשתמש למחוק ספר. כיצד נצליח אנחנו, בעת הצורך, למחוק ספר?

תשובה: אנחנו, מנהלי המערכת, ניכנס ב-User אחר, עם הרשאה מתאימה לביצוע מחיקה. מסד הנתונים יכול לנהל משתמשים רבים. לכל גולשי האינטרנט, יהיה אותו User, עם הרשאות מתאימות ויכולת מוגבלת לעומת ה-User שלנו.

שאלה: ואם הם ישכחו את הסיסמה?

תשובה: הם לא צריכים לזכור את השם או הסיסמה, **ולא צריכים** אפילו לדעת על קיומם. עצם הכניסה לדף ASP שנבנה, גורמת לקישור להתבצע. השם והסיסמה כתובים בתוך דף ASP, כחלק מההגדרת ה-Connection.

קישור משותף – Connection Pooling

ברוב הדפים שנכתוב בספר זה, ניצור קישור למסד נתונים בתחילת הדף, ונסגור אותו בסיום הדף. במהלך הקישור, מבוצעים שלבים רבים הגוזלים משאבי מחשב: בדיקת שם וסיסמה, מתן הרשאות וכו׳. רעיון טוב הוא ליצור קישור פעם אחת, בתחילת התוכנית, ולהשתמש בו בכל פעם שצריך. במילים אחרות: במקום ליצור את הקישור ברמת דף בודד, ניצור אותו ברמת Session ואפילו ברמת Application.

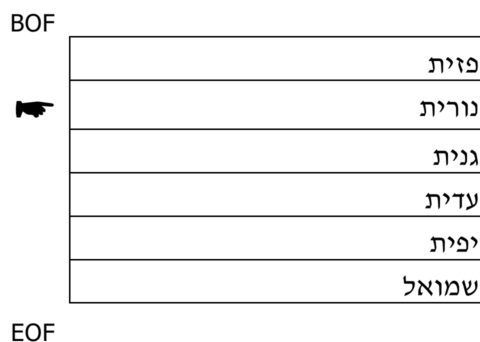
Connection Pooling אינו תכונה של ASP אלא הוא חלק מ- ODBC. הרעיון עובד כך: במקום ליצור קישור עבור כל דף, נשמרים מספר קבוע של קישורים פתוחים למסד הנתונים. כל דף שמבקש קישור – מקבל את אחד הקישורים הפנויים. תאורטית, בשיטה זו אמור להתקבל שיפור ביצועים גדול, ללא כל מאמץ. בפועל, העניינים מעט יותר מורכבים. ניתן להגדיר כלל אצבע בנושא שיטות הקישור:

- אם מספר המשתמשים בו-זמנית באתר קטן – יש שיפור ביצועים משמעותי ב- Connection Pooling. כלומר, כדאי ליצור את הקישור בתוך הדף Global.asa (הסברים על דף זה בפרק מיוחד), ולהשתמש בו בכל מקום שצריך.
- אם מספר המשתמשים בו-זמנית באתר גדול – כדאי יותר ליצור קישור נפרד לכל דף, כמו בדוגמאות בספר זה.

Recordset

המילה העברית הקרובה ביותר ל-Recordset שאני מכיר היא "רשימה". כשמה כן היא: אוסף של רשומות. אם נבקש לקבל את כל טבלת התלמידים בבית הספר, זוהי רשימה. אם נבקש את רשימת התלמידים ששמם מתחיל באות ג, מסודרים לפי מספרי זהות, גם זוהי רשימה. כלומר: רשימה - Recordset, היא אובייקט המסוגל לקבל נתונים ממסד הנתונים, שהם תוצאת שליפה. בחלק מסוגי הרשימה ניתן לטייל על פני הרשימה, ללכת קדימה ואחורה, לעדכן את המידע, להוסיף שורות חדשות ועוד. בסוגים אחרים – לא. אובייקט הרשימה הוא אמצעי עזר לשליפת נתונים ממסד הנתונים וכתיבתם בדפי ASP.

דמיינו את האובייקט Recordset כארון מגירות גדול. בפקודה אחת אנו ממלאים את מגירותיו בערכים, ומרגע זה אנו יכולים לפתוח כל מגירה שנרצה ולקרוא את תוכנה. לארון יש "מצביע" - יד קטנה שמצביעה תמיד על אחת המגירות. ניתן להזיז את המצביע קדימה ואחורה לפי רצוננו. פעולות שנבצע יתבצעו תמיד על השורה עליה מורה המצביע.



בנוסף למגירות רגילות יש לארון שלנו עוד שני מיקומים אפשריים למצביע.

1. **BOF**, Beginning Of File. זהו **מקום אחד לפני הרשומה הראשונה**.

   מצב זה יכול לנבוע מכמה סיבות. האחת: הרשימה פשוט ריקה, בארון שלנו אין אף מגירה. לדוגמה, ביקשנו את כל מתכנתי ASP בחברה, ששכרם פחות מ- 10000 שקל, ואין כאלה. אז המצביע עומד על BOF. סיבה שנייה: מישהו העביר את הסמן אחורה בלולאה והגיע עד ראש הרשימה.

2. **EOF**, End of File. זהו **מקום אחד אחרי הרשומה האחרונה**. הימצאות ב-EOF מעידה כי מישהו העביר את הסמן עד הסוף ועוד אחד, או שהרשימה פשוט ריקה.



טיפ:

השיטה המהירה ביותר לדעת אם רשימה שקיבלתם היא ריקה: אם EOF=true וגם BOF=true.

---

60 שניות על אובייקטים, פעולות ותכונות

בטכנולוגיית האובייקטים משתמשים במונחים **Method** (שיטה) **ו-Property** (מאפיין).

האובייקט הוא חפץ כלשהו בו אנו עוסקים: חתול, ילד, חשבונית, מכונית, ספר, תלמיד.

השיטות הן סך כל הפעולות שהאובייקט יודע לבצע. לדוגמה, האובייקט שנקרא מכונית, יודע לבצע מספר פעולות: לנסוע קדימה, לנסוע אחורה ולצפצף. בשפת האובייקטים נכתוב:

```
mycar.kadima
```

והתגובה תהיה, כך יש לקוות, נסיעה קדימה.

המאפיינים (properties), לעומת זאת, אינם מבצעים שום דבר, הם פשוט קיימים. מאפיינים של מכונית, אם להמשיך בדוגמה הקודמת, הם צבע, דגם ושנת ייצור. את צבע המכונית ניתן לקרוא מתוך האובייקט, להעבירו למשתנה ולהדפיס אותו. לעיתים, אך לא תמיד, ניתן גם לשנות את מאפיין האובייקט בעזרת השמה (קרא משמאל לימין):

מאפיין.אובייקט = **ערך**

שימו לב לנקודה המפרידה בין האובייקט למאפיין.

המשמעות היא: קח את התוכן של המאפיין בתוך האובייקט ואחסן אותו במשתנה בשם ערך.

דוגמאות מעולם הרכב:

```
a = mycar.color
print mycar.color
```

וכך צובעים מכונית שלמה ללא מאמץ:

```
mycar.color = red
```

ביצוע פעולות על Recordset

עכשיו, אחרי הכשרה מעמיקה בטכנולוגיית האובייקטים, כולנו Object Oriented, ומוכנים לקלוט את רשימת המאפיינים והשיטות של האובייקט Recordset.

Method	הסבר
AddNew	הוסף רשומה חדשה
Close	סגור
Delete	מחק רשומה שלמה מהטבלה
Move	שנה את מיקום המצביע
MoveFirst	עבור לרשומה הראשונה
MoveLast	עבור לרשומה האחרונה
MoveNext	עבור רשומה אחת קדימה
MovePrevious	עבור רשומה אחת אחורה
Open	פתח את הרשימה
Requery	פתח את הרשימה מחדש (בדרך כלל לאחר שאחרים ביצעו שינויים)
Update	עדכן את הרשימה (לאחר שאנחנו ביצענו שינויים)

וכעת רשימת המאפיינים:

Property	הסבר
BOF	Bottom of File המצביע עומד לפני הרשומה הראשונה
EOF	End of File המצביע עומד אחרי הרשומה האחרונה
Bookmark	כמו סימניה בספר, שומר על הצבעה למקום מסוים
EditMode	מצב העריכה של הרשימה
Filter	מסנן. שיטה לקבלת רשימה חלקית לפי קריטריונים
LockType	סוג הנעילה המבוצע על הרשומות בזמן עריכה
MaxRecords	קובע מספר מירבי של רשומות שיחזרו מהרשימה
Source	מקור הרשומות: שם טבלה, משפט SQL, וכו'
State	מצב הרשימה: פתוחה או סגורה
Status	מצב הרשומה הנוכחית

Command

הפקודה Command היא אובייקט נוסף הנמצא מתחת ל-Connection, במודל האובייקטים. כלומר – לכל קישור יכולות להיות פקודות רבות. המילה "פקודה" מעט מבלבלת: אין שום קשר לפקודות תכנות. זהו אובייקט המאגד תחתיו מספר הוראות שאנו יכולים לתת למסד הנתונים:

1. בצע שאילתת SQL. במקרה זה, פקודה דומה מאוד לרשימה. התחביר מעט שונה, אבל הרעיון זהה: קח משפט SQL, תן מידע.

2. הפעל Stored Procedure. "פרוצדורה מאוחסנת" היא תוכנית היושבת בשרת מסד הנתונים, וניתן להפעיל אותה בכל פעם שצריך. יתרונה העיקרי: מכיון שהיא בשרת, וכבר עברה הידור (קומפילציה) היא רצה מהר מאוד. Access אינו תומך בפרוצדורות מאוחסנות, SQL Server – כן. נשתמש בפרוצדורות מאוחסנות, לביצוע מטלות חוזרות על עצמן, הקשורות לנתונים.

3. הבא טבלה בשלמותה.

יצירת קישור למסד נתונים

כדי ליצור אובייקט Connection, נשתמש בפעולה CreateObject של האוביקט Server.

```
dim Conn
set Conn=Server.CreateObject("ADODB.Connection")
```

לאחר יצירת אובייקט הקישור, יש לחבר אותו בפועל למסד הנתונים. ניתן לבצע זאת בשתי דרכים: האחת, לציין בתוך הפקודה את כל המידע הדרוש לקישור: מיהו מנהל ההתקן, היכן נמצא בדיוק מסד הנתונים, מהו שם המשתמש והסיסמה הדרושים לחיבור וכן הלאה. דרך שנייה היא, ליצור DSN (Data Source Name), האוגר בתוכו את כל המידע הנזכר למעלה. ואז, מתוך התוכנית שלנו, נציין רק את שם ה-DSN והנתונים יילקחו ממנו. בדרך כלל, נוח להשתמש ב- DSN. נראה את שתי הדרכים:

ראשונה, הדרך המפורשת. המבנה העקרוני הוא:

Conn.Open *ConnectionString, User, Password*

דוגמה לחיבור למסד נתונים Access:

```
Dim ConnString
ConnString = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & _
                Server.MapPath("bookstore.mdb") & ";"

Conn.Open ConnString
```

הסבר קצר על הפקודה Server.MapPath. זו פקודה שממפה מיקום פיסי למיקום יחסי לשרת. זה משפט קשה, ולכן אחזור עליו במילים אחרות. אם תנסו לכתוב בתוך שורת הקישור:

```
Data Source="c:\bookstore\bookstore.mdb"
```

תקבלו מייד הודעת שגיאה. מיקום בסיס הנתונים לא יכול להיות מסלול פיסי, כמו בדוגמה, אלא מסלול שמתחיל בספריה הווירטואלית של הפרויקט בשרת. אם הספריה הזו היא c:\inetpub\wwwroot\bookstore, אז אם אני נמצא בספריה זו, לאן עלי ללכת **מכאן** כדי למצוא את מסד הנתונים? המקרה שלנו הוא טריוויאלי, הקובץ נמצא בדיוק

בספריה זו, ולכן התשובה: אל תלך לשום מקום, הנה הוא כאן לפניך ושמו bookstore.mdb. ובעברית:

```
Data Source=Server.MapPath("bookstore.mdb")
```

ואם הייתי רוצה להיות מסודר ולפתוח ספריה מיוחדת, נאמר בשם db, שבה יהיה מסד הנתונים? אז המסלול הפיסי של הקובץ הוא:

```
c:\inetpub\wwwroot\bookstore\db\bookstore.mdb
```

מה יהיו הוראות הגישה למסד הנתונים במקרה זה? מנקודת המוצא שלך, לך לספריה db, ובה תמצא קובץ בשם bookstore.mdb. הנה כך:

```
Data Source=Server.MapPath("db\bookstore.mdb")
```

זו השיטה בה צריך לעבוד בשרתים בהם **לא** ניתן לייצר DSN, ויש רבים כאלה. רוב שרתי החינם לא מאפשרים יצירת DSN, ולכן כדאי לדעת גם כיצד להסתדר בלעדיו.

להלן מספר קודים להתקשרות למסדי נתונים שונים:

קישור ל-Access תוך שימוש ב-ODBC:

```
oConn.Open "Driver={Microsoft Access Driver (*.mdb)};" & _
"Dbq=\somepath\mydb.mdb;" & _
"Uid=Admin;" & _
"Pwd=;"
```

קישור ל-Access תוך שימוש במנוע Jet:

```
Conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source=\Databases\bookstore.mdb;" & _
"User Id=admin;" & _
"Password=;"
```

קישור ל- SQL Server תוך שימוש ב-ODBC:

```
Conn.Open "Driver={SQL Server};" & _
"Server=MyServer;" & _
"Database=Sifroni;" & _
"Uid=sa;" & _
"Pwd=;"
```

קישור ל- SQL Server תוך שימוש ב- OLE-DB:

```
oConn.Open "Provider=sqloledb;" & _
"Data Source= MyServer;" & _
"Initial Catalog= Sifroni;" & _
"User Id=sa;" & _
"Password=;"
```

הגדרת DSN

קישור ל- Oracle תוך שימוש ב- OLE-DB :

```
Conn.Open "Provider=msdaora;" & _
"Data Source=OracleServer.world;" & _
"User Id=david;" & _
"Password=;"
```

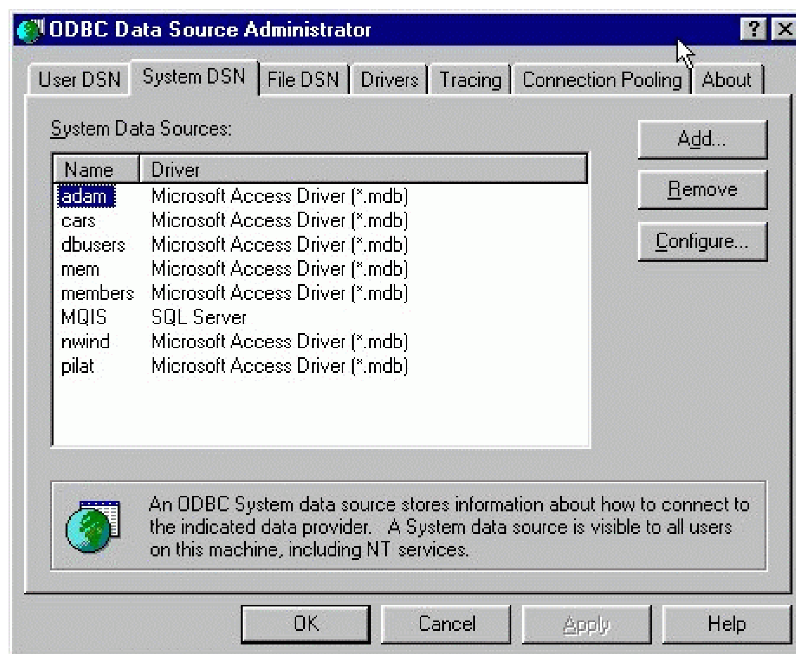
כן, גם Excel יכול לשמש כמקור מידע :

```
Conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source=\MyExcellFiles\customers.xls;" & _
"Extended Properties=""Excel 8.0;HDR=Yes;"";"
```

כעת, לדרך הקלה, וגם המקובלת יותר, של DSN. תחילה, נגדיר DSN חדש. עושים זאת בשרת, מפני שהוא זה שמריץ את תוכניות ASP, לכן מיקום מסד הנתונים ביחס אליו הוא הקובע. אם אתם מפתחים במחשב מקומי, השרת הוא המחשב שלכם, ולכן אין זה משנה.

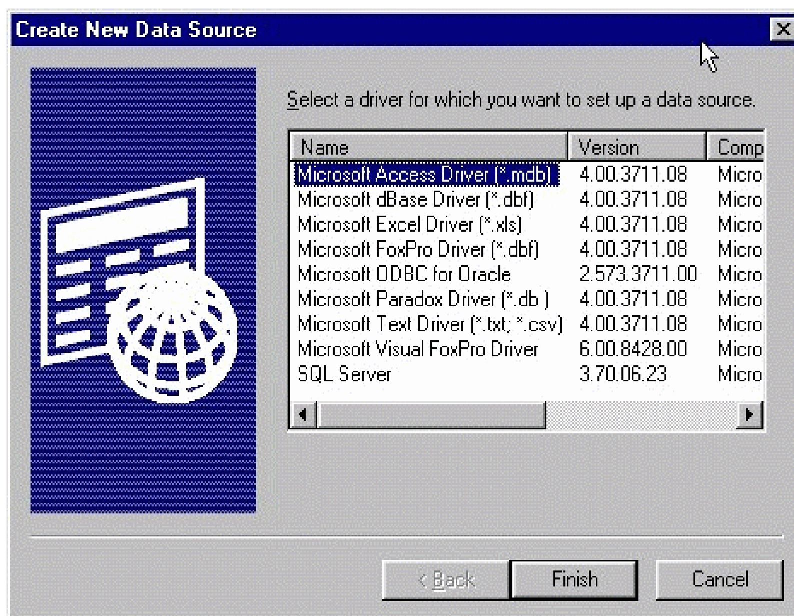
בלוח הבקרה (Control Panel) בחרו ODBC Data Sources (אם אתם משתמשים ב-Windows 2000 Professional, בחרו בלוח הבקרה ב- Administrative Tools ושם תמצאו את (Data Sources (ODBC). בחלון הבא בחרו System DSN. אל תשתמשו ב-File-DSN. ההבדל : ב- File DSN נשמר המידע בקובץ טקסט רגיל בדיסק, וב- System DSN המידע נשמר ב- Registry. הגישה לנתונים דרך System DSN מהירה לפחות פי שלושה מאשר ב- File DSN.

User DSN יפעל רק עבור המשתמש שיצר אותו, לכן הוא אינו מתאים לנו.



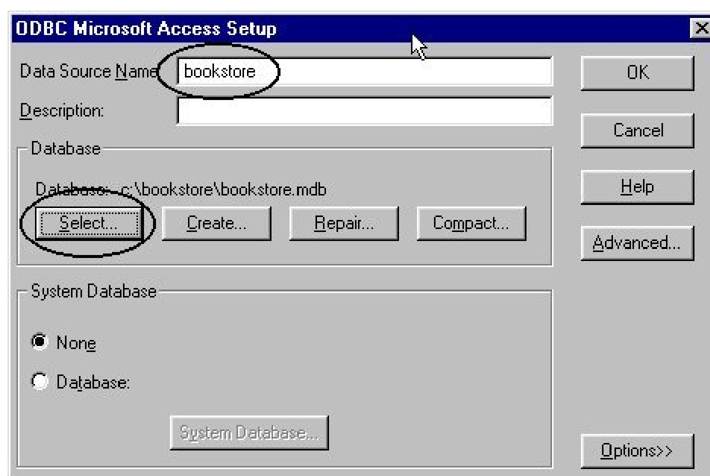
כדי להוסיף System DSN, לחצו על הלחצן Add.

יתקבל החלון הבא:



זוהי רשימה של כל ה- ODBC Drivers, המותקנים במחשב שלי. אצלכם הרשימה עשויה להיות שונה. מנהלי ההתקן ברשימה המופיעה למעלה, הותקנו על ידי Office, אך הם אופציונליים. כלומר, בעת התקנת Office, ניתן להתקינם או לא. אם חסר לכם אחד מהם, היכנסו לתוכנית ההתקנה של Office, בחרו ב- Data Access, וסמנו את מנהלי ההתקן החסרים.

נבחר במנהל ההתקן של Access ונמשיך.



בחלון זה יש לבצע שתי פעולות:

1. לתת שם ל- DSN. זה קל. בדרך כלל ניתן לו שם זהה לשם מסד הנתונים. כך נחסוך תאי זיכרון במוח.
2. לבחור במסד הנתונים. לחצו על Select. ייפתח חלון רגיל של פתיחת קובץ. מצאו את קובץ mdb שלכם, ואשרו. כדאי שמסד הנתונים יהיה סגור בעת ביצוע פעולה זו, אחרת עלולה להופיע הודעת שגיאה.



הערה:
חלון זה הוא ספציפי ל- Access! עבור מסדי נתונים אחרים, יהיו חלונות אחרים, עם שאלות המתאימות להם.

לחצו על OK וחזרו ל- ODBC Data Source Administrator. ודאו שה- DSN החדש שלכם מופיע ברשימת System DSN.



הערה:
בדוגמה זו, ובמקומות אחרים, שם ה- DSN זהה לשם קובץ מסד הנתונים. אין זו חובה. השמות יכולים להיות שונים לגמרי. כדי שיהיה לנו פחות מה לזכור, כדאי לתת את אותו שם.

פינת הגאון - יצירת DSN
1. לוח בקרה -> ODBC
2. בחר System DSN, **ולא שום דבר אחר!** לחץ Add.
3. בחר בסוג מסד הנתונים הנכון, ולחץ Finish.
4. תן שם ל- DSN החדש, ובחר בלחצן Select את מסד הנתונים.

יצירת קישור בעזרת DSN

כעת, שיש לנו DSN חדש ויפה, ניצור בעזרתו קישור למסד הנתונים. להלן התחביר:

Conn.Open *ConnectionString, User, Password*

מימוש פשוט:

```
Conn.Open "bookstore"
```

מימוש מורכב יותר:

```
Conn.Open "DATABASE=bookstore;DSN=bookstore;UID=sa;Password=;"
```

לאחר סיום העבודה עם האובייקט Connection, יש לסגור אותו ולשחרר את הזיכרון שתפס:

```
Conn.Close
Set Conn=Nothing
```

ביצוע פעולות בעזרת Connection

בעזרת האובייקט Connection, ניתן לבצע פעולות על מסד הנתונים. פעולות כאלו יהיו שאילתות SQL, Stored Procedures (שאילתות מאוחסנות בשרת, תכונה שאינה קיימת ב-Access) וכן הלאה. להלן התחביר:

Conn.Execute *Command*

לדוגמה:

```
Conn.Execute "insert into users (fname, lname) select 'moshe', 'cohen' "
```

זוהי שאילתת הוספה, המוסיפה רשומה לטבלת users. כאמור, ספר זה אינו עוסק בשפת SQL, אך אל דאגה, השאילתות שנבצע יהיו בדרך כלל פשוטות מאוד.

לסיכום נושא Connection, דוגמה נוספת, והפעם כל הקוד הנדרש, כולל הגדרת הקישור:

```
dim Conn
Set Conn=Server.CreateObject("ADODB.Connection")
Conn.Open "mydsn"
Conn.Execute "delete from customers where last_visit < #1/1/2000#"

Conn.Close
Set Conn=Nothing
```

הסבר:

- המצאנו אובייקט חדש.
- הודענו לו שהוא מעכשיו אובייקט מסוג Connection.
- חיברנו אותו ל- DSN, שידע למי להתקשר, וכיצד.
- ביצענו פעולה. מחקנו את כל הלקוחות, שתאריך הביקור האחרון שלהם באתרנו קטן מ...
- סגרנו את האובייקט מסוג Connection.
- פינינו את מרחב הזיכרון שתפס האובייקט.

פתיחת Recordset

ניתן לחלק את הפעולות המבוצעות מול מסד הנתונים לשתי קבוצות:

1. שאילתות ביצוע (Action Queries). תפקידן לבצע פעולה כלשהי: מחיקת רשומה, הוספת רשומה, עדכון רשומה וכדומה. פעולות אלו אינן מחזירות רשומות.
2. שאילתות בחירה (Select Queries). שאילתות שמטרתן לקבל נתונים ממסד הנתונים. אלו יכולות להיות פשוטות או מורכבות, אך תמיד יחזירו תוצאה כלשהי במבנה טבלאי.

שאילתה לדוגמה: הראה לי את כל העובדים ממחלקה 3, ממוינים לפי שם משפחה:

```
Select * from employees where department_id=3 order by fname
```

האובייקט המכיל רשומות נקרא Recordset, והוא כלי הקיבול שלתוכו נשפוך את תוצאות השאילתה. עליו גם נבצע פעולות לאחר מכן: נשנה נתונים, נקרא אותם לתוך טבלה למסך, נסרוק אותם בלולאה וכן הלאה.

תחילה, יש ליצור את אובייקט Recordset:

```
dim rs
set rs = Server.CreateObject("ADODB.Recordset")
```

לאחר מכן יש לפתוח אותו. לפתיחה דרושים לפחות שני פרמטרים:

1. מה לפתוח? זו יכולה להיות שאילתת SQL, טבלה שלמה, או Stored Procedure.
2. חיבור פועל למסד הנתונים, ובעברית - Connection.

יתר הפרמטרים אופציונליים. להלן התחביר לפתיחת Recordset:

Recordset.Open Source, Connection, CursorType, LockType, Options

לדוגמה:

```
dim conn, rs, SQL
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open "dsn=mydatabase"
sql="select * from orders where total_sum > 150"
Set rs=Server.CreateObject("ADODB.Recordset")
rs.Open sql, conn
```

הסבר:

רצינו לקבל את רשימת ההזמנות שסכומן מעל 150 שקלים. יצרנו אובייקט מסוג connection, קישרנו אותו ל- DSN, שנקרא mydatabase. לתוך המשתנה SQL הכנסנו את השאילתה שלנו. יצרנו אובייקט מסוג Recordset ובשורת המחץ, פתחנו אותו עם שני הפרמטרים: שאילתה וקישור.

חוקי גם התחביר הבא:

Set *Recordset = Connection.*Execute*(RecordsAffected, Parameters, Options)*

לדוגמה:

```
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open "mydatabase"
set rs = conn.Execute("select * from orders where total_sum < 150")
```

פינת הגאון - יצירת Recordset

```
dim conn, rs, sql
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open "dsn=mydatabase"
sql="select * from orders where total_sum > 150"
Set rs=Server.CreateObject("ADODB.Recordset")
rs.Open sql, conn

     כאן תבצע פעולות על Recordset. . . . . . . .

rs.Close
set rs=Nothing
conn.Close
set conn=Nothing
```

דוגמה לשימוש ב-Connection ו-Recordset

ננצל את הידע שרכשנו לבניית הדף הראשון בפרויקט שלנו, דף רשימת הספרים בשם booklist.asp:

```
<table border=1>
<%
dim conn, rs, sql
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open "dsn=bookstore"
sql="select * from books"
Set rs=Server.CreateObject("ADODB.Recordset")
rs.Open sql, conn

do while not rs.eof
%>
<tr><td><%=rs("title")%></td></tr>
```

```
<%
 rs.MoveNext
loop

rs.Close
set rs=nothing
conn.Close
set conn=nothing
%>

</table>
```

הסבר:

יצרנו קישור ופתחנו recordset לטבלת הספרים. כעת אנו יוצרים טבלת html, ועבור כל שורה בטבלת הספרים, מוסיפים שורה בטבלת ה-html. שימו לב למעברים בין עולם ה-html לעולם ה-asp: הגדרת הטבלה (<table>) היתה html טהור. עברנו ל-asp כדי ליצור קישור, ולכן מופיע הסימן %>. באמצע הלולאה נזקקנו לחזור ל-html ולכן סגרנו את הקוד ב- <% וכך הלאה. נסו זאת בעצמכם, והריצו. התוצאה צריכה להיות טבלה ובה שמות כל הספרים שלכם. נסו לשכלל את הטבלה, לצבוע בצבעים שונים, להוסיף כותרות, ולהוסיף עמודות נוספות לטבלה.

כדי לשכלל את הדוגמה ולהפוך אותה למקצועית יותר, נלמד פקודה חדשה. נשתמש באובייקט Response כדי לכתוב טקסט רגיל למסמך. יכולת זו תמנע מאיתנו את הצורך לעבור לעיתים תכופות בין עולם ה-ASP לעולם ה-HTML. התחביר:

```
Response.Write Text
```

כלומר, בדוגמה שלנו, במקום כל הקטע:

```
<tr><td><%=rs("title")%></td></tr>
```

נכתוב שורה בודדת:

```
Response.Write "<tr><td>" & rs("title") & "</td></tr>"
```

אין צורך עוד במעבר ל-HTML מפני שהשורה שלנו כולה ASP! הפקודה כותבת את הטקסט שנמסר לה כמו שהוא, ישר לדף התוצאה, לדפדפן של הגולש. במקרה שלנו, זהו טקסט משורשר. כל מה שבמרכאות - מצוטט במדויק. הביטוי **rs("title")** מכיל שם ספר כלשהו (אחר בכל שורה!) והוא יישלח לדף התוצאה.

לסיכום, להלן הקובץ booklist.asp. הקלידו אותו וודאו שהוא עובד היטב (קוד המקור נמצא בקובץ booklist2.asp):

```
<%@ Language=VBScript %>

<%
dim conn
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"

dim rs, sql
set rs=Server.CreateObject("ADODB.recordset")
sql="select * from books order by book_id"
rs.Open sql, conn

Response.Write "<center><table border=1 cellpadding=2 BGCOLOR=Yellow>"
do while not rs.EOF
 Response.Write "<tr>"
 Response.Write "<td align=right>" & formatcurrency(rs("price")) & "</td>"
 Response.Write "<td align=right>" & rs("author") & "</td>"
 Response.Write "<td align=right>" & rs("title") & "</td>"
 Response.Write "</tr>"
 rs.MoveNext
loop

Response.Write "</center></table>"
rs.Close
set rs=nothing
conn.Close
set conn=nothing
%>
```

הערות והסברים:

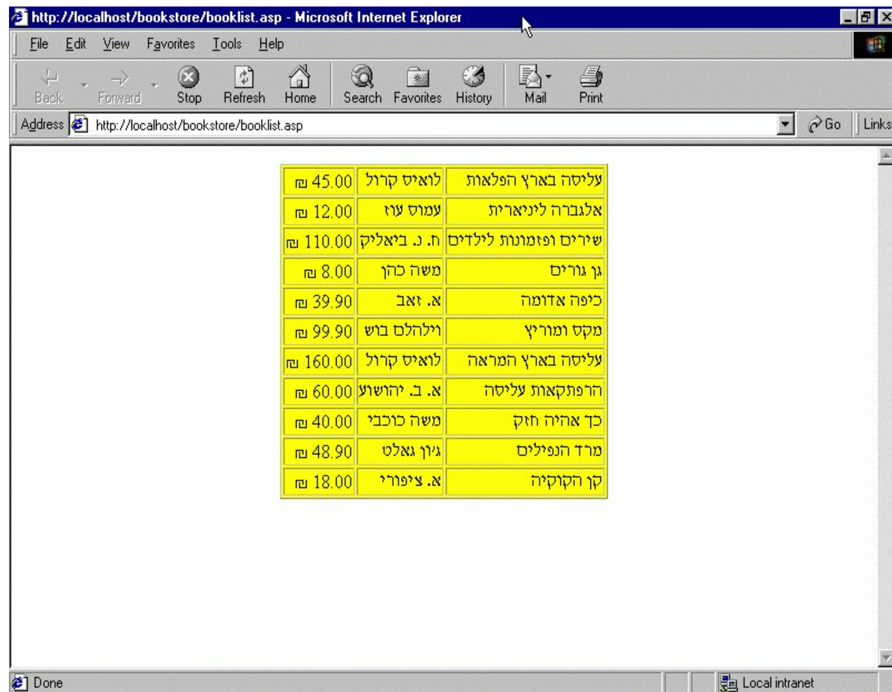
פתחנו קישור ורשימת נתונים (Connection, Recordset), כפי שנאמר והוסבר.

לאחר מכן, פתחנו בלולאת סריקה על פני כל הרשומות. כל עוד לא הגעתם לסוף הרשימה (eof) - המשיכו קדימה. עבור כל שורה אנו כותבים <TR> כדרוש, ועבור כל עמודה <TD> ואת המידע מתוך הרשימה.

שימו לב, בשורה המטפלת במחיר הספר, לפונקציה FormatCurrency. הפונקציה מסדרת את המספר בפורמט של סכומי כסף, ומצמידה לו את סמל המטבע (איזה?).

שימו לב גם לסדר כתיבתם של השדות. סדר השדות יהיה משמאל לימין, גם אם הם עבריים, לכן את השדה השמאלי ביותר כותבים ראשון.

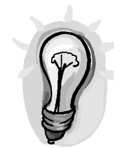
זו צריכה להיות, פחות או יותר, התוצאה בדפדפן:



עליסה בארץ הפלאות	לואיס קרול	45.00 ₪
אלגברה ליניארית	עמוס עוז	12.00 ₪
שירים ופזמונות לילדים	ח. נ. ביאליק	110.00 ₪
גן גורים	משה כהן	8.00 ₪
כיפה אדומה	א. זאב	39.90 ₪
מקס ומוריץ	וילהלם בוש	99.90 ₪
עליסה בארץ המראה	לואיס קרול	160.00 ₪
הרפתקאות עליסה	א. ב. יהושוע	60.00 ₪
כך אהיה חזק	משה כוכבי	40.00 ₪
מרד הנפילים	ג'ון גאלט	48.90 ₪
קן הקוקיה	א. ציפורי	18.00 ₪

יפה?

מי שזקוק להוכחה נוספת, יכול להוסיף ספר לטבלה, או למחוק ספר, להריץ את הדף שוב, ולראות את השינוי מתבטא בדף.



טיפ:

באקסס 2000 יש פטנט נחמד שכדאי לדעת: יצירה אוטומטית של דף ASP מטבלה או שאילתה. לעיתים אתם צריכים להראות טבלה שלמה, או שאילתה, כמו שהיא, בדף אינטרנט. סמנו את הטבלה או השאילתה, בתפריט קובץ בחרו "יצא..." ונפתח חלון שמירה. בתיבה "שמור כסוג" בחרו ASP. נפתח עוד חלון קטן עם כמה שאלות כמו למשל "שם מקור הנתונים", שזה בעברית DSN. אשרו, ויש לכם דף ASP מוכן לשימוש. בהרבה מקרים תצטרכו לשפר אותו מעט, בעיקר את הצד העיצובי, אבל זהו ללא ספק חיסכון ניכר בזמן עבודה.

חלוקה לדפים

כל עוד תוצאות החיפוש יניבו עד 10 או 20 ספרים, הדף booklist.asp יעבוד יפה. אך מה יקרה אם יהיו 400 ספרים שיתאימו לדרישה? יקרו כמה דברים "רעים":

1. הדף יעלה מאוד לאט.
2. הוא יהיה מאוד לא נוח לצפייה.
3. הוא יצרוך הרבה מאוד זיכרון מהשרת.

הפתרון: שימוש ב-Pages. **לפני** פתיחת הרשימה, קבעו את גודל הדף:

```
rs.PageSize=20
```

מעתה, כל דף יכיל 20 רשומות, זה גודל סביר. כעת נפתח את הרשימה בשיטה הרגילה, והיא תכיל את הדף הראשון, כלומר את 20 הרשומות הראשונות. נתייחס אליה כאל רשימה רגילה, ונציג את הרשומות בטבלה, או בכל דרך אחרת. כמה דפים כאלה יש? אנחנו הרי רוצים לכתוב למשתמש: דף 1 מתוך 14. ובכן, את מספר הדפים נקבל בעזרת המתודה PageCount. הנה כך:

```
x=rs.PageCount
```

כיצד נקפוץ לדף מסוים? הנה כך:

```
rs.AbsolutePage=7
```

כך נעבור לדף מספר 7. מובן שיש צורך שוב לכתוב לדף את תוצאת השאילתה.

עבודה עם ADO

טכנולוגיית ADO היא אוסף של אובייקטים המיועדים לפשט את העבודה מול מסדי נתונים. בספר זה רוב הדוגמאות יהיו תוך שימוש בשפת SQL, שיש לה יתרונות. אך כדאי להכיר גם את שיטות העבודה של ADO, בעיקר בזכות הקלות המופלאה שלהן. למען האמת, כבר השתמשנו רבות ב-ADO: האובייקטים Connection ו-Recordset שייכים למודל ADO.

הוספת שורה לטבלה בעזרת ADO

```
rs.Open "customers", "dsn=bookstore", 2, 3, 2
rs.AddNew
rs("name")="Yosi"
rs("phone")="03-1234567"
rs("email")="yosi@blabla.com"
rs.Update
```

שינוי פרטי רשומה בעזרת ADO

```
Sql="select * form customers where cust_id=17"
rs.Open sql, "dsn=bookstore", 2, 3, 2
rs.Edit
rs("name")="Moshiko"
rs("phone")="03-7654321"
rs("email")="moshiko@blabla.com"
rs.Update
```

זה פשוט, זה אינטואיטיבי, וקל מאוד לאתר פה שגיאות. אם חושדים בשדה מסוים, מוחקים או הופכים לשורת הערה את השורה הבעייתית, ומנסים בלעדיה.

מהם המספרים המשונים שבאים אחרי פקודת פתיחת הרשימה, 2,3,2? אלה פרמטרים שונים שמורים לשרת באיזו שיטה לפתוח את הרשימה. נדון בהם מאוחר יותר בספר.

בתקליטור המצורף לספר תמצאו דוגמה פשוטה ביותר לשימוש ב-ADO. הדף ado1.asp קולט פרטי ספר חדש, ומעביר אותם לדף שני, ado2.asp. דף זה, בעזרת ADO, מכניס את הנתונים לטבלה.

הקובץ adovbs.inc

כאן גם המקום להזכיר את הקובץ adovbs.inc. זהו קובץ טקסט פשוט, שמכיל הגדרות. כלומר: תרגום של שמות למספרים. נציץ לקטע קטן מתוך הקובץ:

```
'---- CursorTypeEnum Values ----
Const adOpenForwardOnly = 0
Const adOpenKeyset = 1
Const adOpenDynamic = 2
Const adOpenStatic = 3
```

מוגדרים כאן קבועים, Constants, ולכל אחד יש ערך. אם נכלול (include) את הקובץ הנ"ל בדפים שלנו, נוכל לכתוב adOpenStatic במקום 3. זה טוב בעיקר משתי סיבות, האחת: הקוד יהיה יותר קריא וברור. השנייה, כשאתם כותבים פקודת Open לרשימה, Visual InterDev עוזר לכם ומציג את רשימת האפשרויות לכל פרמטר. כמו בדוגמה זו:

```
<%@ Language=VBScript %>

<%

dim keyword
keyword=Request.Form("keyword")

dim conn
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"

dim rs, sql
set rs=Server.CreateObject("
sql="select * from books whe                     keyword & "%'"
rs.Open sql, conn,adOpenStatic,
```

```
adLockBatchOptimistic
adLockOptimistic
adLockPessimistic
adLockReadOnly
adLockUnspecified
```

```
Open (Source, ActiveConnection, [CursorType], [LockType], [Options])
```

אנחנו מוזמנים לבחור את סוג הפתיחה וסוג הנעילה מתוך הרשימה, אך כשנריץ את הקובץ תתקבל הודעת שגיאה, מפני שמנוע ASP אינו יודע מה זה adOpenStatic. מבחינתו הסוגים הלגיטימיים הם המספרים 0 עד 3. הקובץ adovbs.inc יוצר עבורנו תרגום: adOpenStatic שווה 3.

הקובץ נמצא בתקליטור המצורף לספר, יחד עם כל קבצי המקור, אך סביר להניח שהוא נמצא כבר במחשב שלכם כחלק מהתקנת שרת האינטרנט. כדי להשתמש בו כתבו בראש הדף את השורה:

```
<!--#include virtual="/adovbs.inc"-->
```

פרק 6

חיפוש על פי מילת מפתח

כעת, בואו נלך עוד צעד אחד קדימה, ונבנה חלון חיפוש, בו נקיש מילה ונקבל את רשימת כל הספרים המכילים מילה זו. המטרה היא כמובן ליצור דף חיפוש בו נקיש מילת מפתח, להגיע ממנו לרשימת תוצאות, ומהרשימה להגיע לדף פרטי הספר המלא.

דף החיפוש שלנו נקרא Searchbook.asp. יצרנו אותו בתחילת הספר, ועכשיו הזמן למלא אותו בתוכן (אם לא יצרתם, צרו כעת דף חדש בשם זה), התוכן יהיה דומה מאוד לדף post1.asp, שכתבנו לפני מספר עמודים, ואפילו פשוט יותר, שכן הוא יכיל רק תיבת טקסט אחת, לחצן אישור (submit) ולחצן ניקוי (reset). דף המטרה שלו (action) יהיה רשימת הספרים שזה עתה סיימנו לכתוב: booklist.asp.

לנועזים: נסו לעשות זאת בעצמכם! בנו קובץ בשם searchbook.asp (קוד המקור נמצא בקובץ searchbook1.asp):

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>
<form name="booksearch" action="booklist.asp" method="post">

<center>
<INPUT id=text1 name=keyword style="HEIGHT: 22px; WIDTH: 175px"><FONT
face=Aharoni
size=5><STRONG>הקש מילה לחיפוש

<BR><BR><BR>

<INPUT type="reset" value="Reset" id=reset1 name=reset1 style="HEIGHT: 24px;
WIDTH: 94px"></STRONG></FONT>  
```

```
<INPUT type="submit" value="Submit" id=submit1 name=submit1 style="HEIGHT:
24px; WIDTH: 100px">

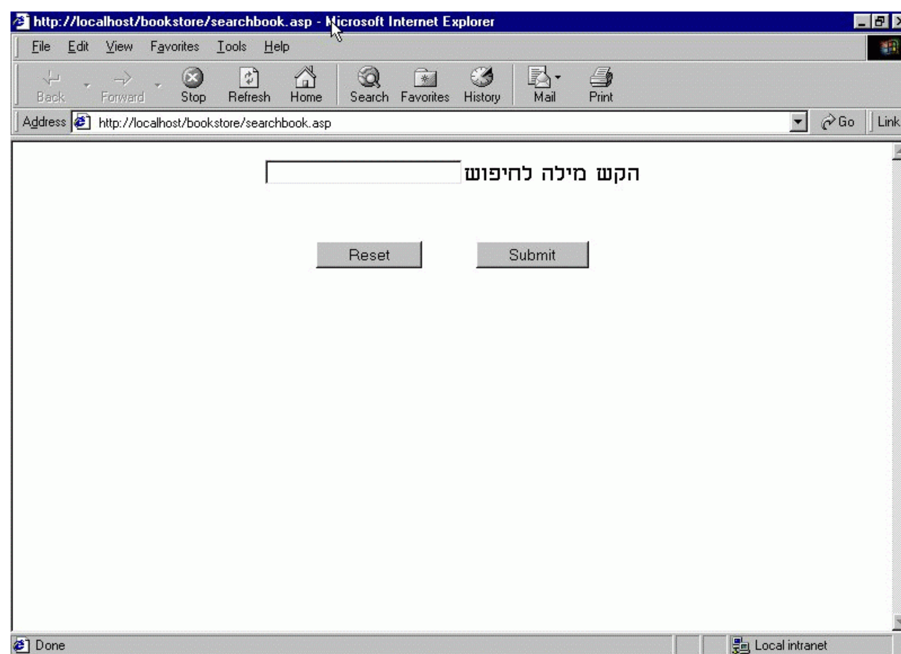
</center>

</form>
</BODY>
</HTML>
```

שימו לב לנקודות הבאות:

1. שם תיבת הטקסט הוא keyword. ניתן לבחור בכל שם, אך יש לו חשיבות. לפי שם זה נקבל את הערך המוחזר בדף הבא.
2. שורת הגדרת הטופס חשובה. במיוחד שימו לב להפניה לדף booklist.asp.
3. המלל העברי נכתב לכם הפוך? לא נורא. בדף HTML הוא ייראה נכון.

כשתריצו, הדף צריך להיראות בערך כך:



לחיצה על לחצן Submit תביא את דף רשימת הספרים, כל הספרים. למה מגיעים כל הספרים ולא רק אלה המכילים את מילת המפתח? מפני שעוד לא כתבנו את זה.

לפני שנפעיל את החיפוש עצמו, נסחט מתוך פרוטוקול http את המילה המבוקשת, ונוודא זאת בדף היעד.

היכנסו לדף booklist.asp (הכי קל - מתוך דיאגרמת האתר), שתלו את הטקסט הבא מייד לאחר התג <body> ולאחר ה- %>. הכוונה היא, שהוא יהיה בתוך עולם ה- ASP, אך לפני פתיחת הקישור.

```
dim keyword
keyword=Request.Form("keyword")

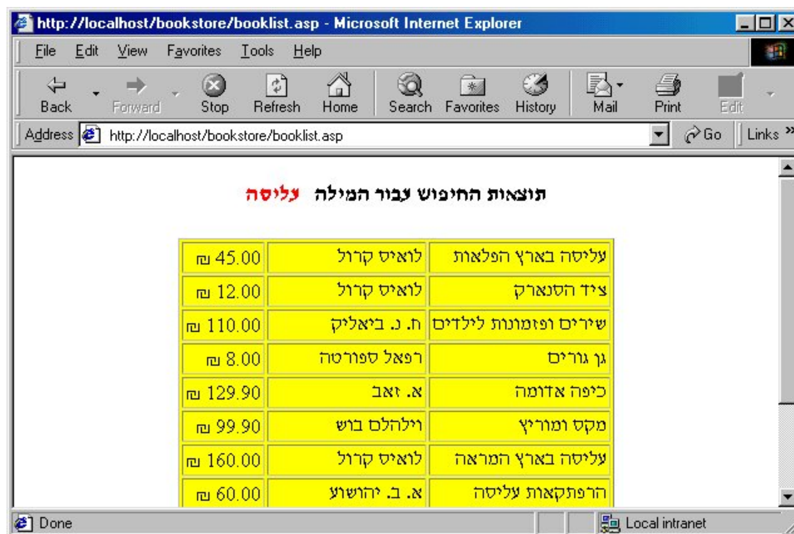
Response.Write "<center><table cellpadding=5><tr>"
Response.Write "<td><h3><FONT color=red>" & keyword & "</FONT></h3></td>"
Response.Write "<td><h3>" & "תוצאות החיפוש עבור המילה" & "</h3></td>"
Response.Write "</tr></table>"
```

הסבר:

קלטו את המשתנה שנשלח אלינו מהטופס שקרא לנו, והציגו אותו בתוך המשפט.

כדי להבטיח שהמשפט "תוצאות החיפוש..." יהיה אכן בצד ימין, והמילה המבוקשת בצד שמאל, אני בונה טבלה וממקם כל משפט במקומו. כך אני נמנע מהפתעות לא נעימות הקשורות לסוגי דפדפנים וגרסאות שונות.

כך נראית התוצאה:



אם לא עובד לכם, בדקו:

1. שהשיטה שבחרתם היא אכן post.
2. ששם המשתנה אותו אתם מחפשים, הוא אכן שמה של תיבת הטקסט בחלון החיפוש.
3. שהתגים <form>, </form> מקיפים את כל רכיבי הקלט.

4. שגיאות כתיב. השוו לטקסט בספר.

שגיאות אחרות - בדקו בפרק **איתור תקלות**.

הגולש הקליד מילה ואנו רוצים למצוא את כל הספרים ששמותיהם מכילים את המילה המבוקשת. הדרך לבצע זאת היא באמצעות שאילתת SQL. אנו נשתמש באופרטור Like.

בדף booklist.asp כבר יש לנו שאילתת SQL, אלא שהיא מביאה את כל הספרים, ויש לשנות אותה. השאילתה המקורית היתה:

```
sql="select * from books"
```

אם מילת המפתח היתה abcd, לדוגמה, אז שאילתת SQL צריכה להיות:

```
select * from books where title like '%abcd%'
```

שימו לב לגרשים המקיפים את המחרוזת: גרש, אחוז, המילה המבוקשת, אחוז, גרש.

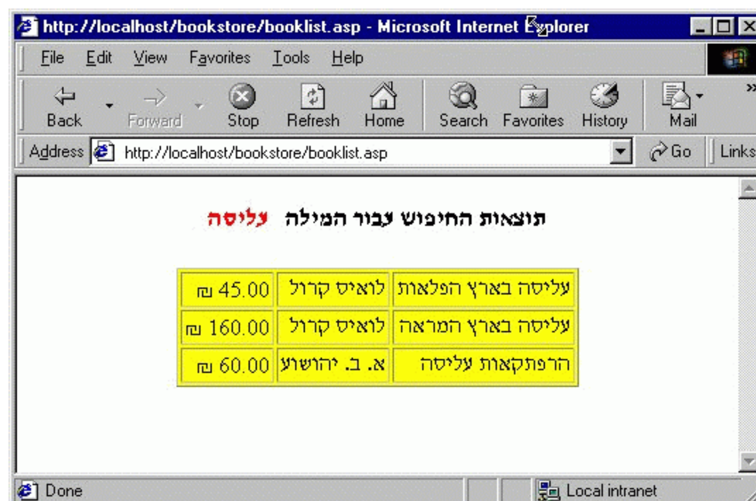
את כל זאת אנו צריכים לבנות באופן דינמי, ולהכניס לתוך המשתנה sql. אנו מצמידים 3 מחרוזות זו לזו בעזרת הסימן &, שהוא "דבק מחרוזות". הנה שלוש המחרוזות:

select * from books where title like '%	keyword	%'

כאשר keyword הוא כמובן המשתנה המכיל את מילת המפתח. הצמידו אותן יחד ותקבלו:

```
sql="select * from books where title like '%" & keyword & "%' "
```

הקלידו שורה זו **במקום** השורה המקורית, והקפידו לא לשכוח אף גרש. רווח לפני ואחרי כל &. שמרו את עבודתכם, עברו לחלון searchbook.asp, הפעילו אותו, הקלידו מילת מפתח, והקישו Submit. צריכה להיפתח רשימת הספרים, ובה רק ספרים ששמם מכיל את המילה המבוקשת.



חזרו לדף החיפוש, הקלידו מילה אחרת, ובצעו חיפוש. ודאו שהתוכנית שלכם עובדת כמו שצריך. אני מקווה שאתם נהנים מהתוצאות.

פתיחת חלון מתוך רשימה, עם פרמטר

רשימת הספרים שפתחנו נחמדה מאוד, אבל כעת רוצה הגולש להיכנס לחלון הפרטים של ספר אחד מסוים. לראות את פרטי הספר, את מחירו, את התמונה שלו, ואולי אף להזמין אותו.

ברור שלא נבנה חלון לכל ספר, אלא חלון אחד שמקבל פרמטר. מה יהיה הפרמטר? כמובן, קוד הספר, book_id. את חלון פרטי הספר נבנה בהמשך. בינתיים נוסיף לחלון הרשימה את האפשרות להגיע לדף זה עם קוד הספר הדרוש. פתחו את הדף booklist.asp מתוך הדיאגרמה, כרגיל, או מתוך ה- Project Explorer.

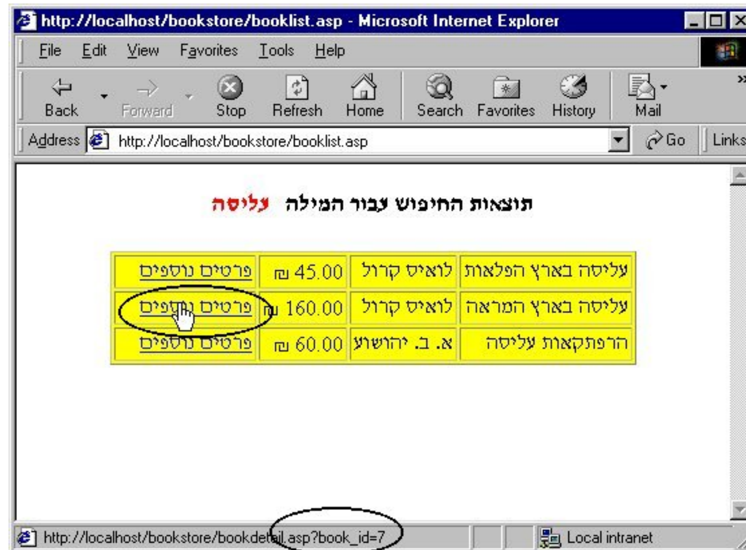
אנו מוסיפים כעת עמודה לטבלה. אם אתם רוצים שהעמודה "פרטים נוספים" תהיה השמאלית ביותר (הגיוני עבור עברית), הוסיפו את העמודה החדשה ראשונה, מייד לאחר התג <tr>. זאת מכיון שהטבלה נכתבת בפועל משמאל לימין, גם אם תוכנה עברי. אלה שורות הקסם שיש להוסיף לדף בתוך לולאת Do While לאחר המשפט Respone.Write "<tr>" (קוד המקור כולו נמצא ב- booklist1.asp):

```
Response.Write "<td width=100 align=right>"
Response.Write "<A HREF=" & "bookdetail.asp?book_id=" & rs("book_id") & ">"
Response.Write "פרטים נוספים" & "</A></td>"
```

למען הסר ספק, שורות אלו מייצרות שורה בטבלה, ולכן צריכות להיכתב מייד לאחר:

```
do while not rs.eof
Response.Write "<tr>"
```

הריצו את החלון, והתוצאה תיראה בערך כך :



שימו לב לכך, שהסמן משנה את צורתו ל״יד״ המצביעה על קישור. כאשר העכבר נמצא מעל השדה החדש. העבירו את העכבר לאורך הטבלה, בעמודת פרטים נוספים, וראו כיצד משתנה הכיתוב בשורה התחתונה, שורת הסטטוס של הדפדפן. בכל שורה הפרמטר הנשלח, הוא קוד הספר הנוכחי.

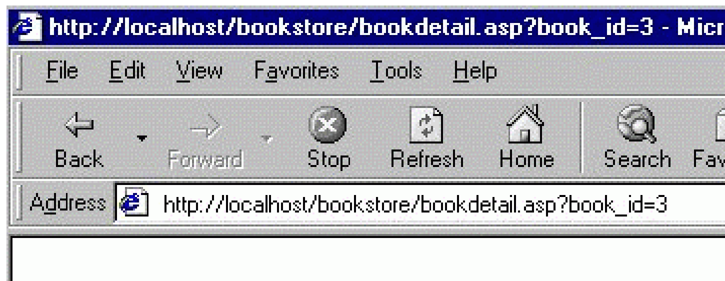
בניית דף הספר הבודד

כעת נבנה את דף פרטי הספר. זהו הדף הסופי בשרשרת החיפוש. בדף זה ימצא הגולש את כל הפרטים על הספר : את תמונתו, מחירו, תקציר וכן הלאה. כאן יהיה הלחצן החשוב ״הוסף לסל הקניות״, ואנו רוצים שילחצו עליו ככל האפשר יותר.

הדף bookdetail.asp כבר קיים. ואם לא - אז ניצור אותו.

קוד המקור לשלב זה נמצא ב- bookdetail1.asp. תחילה, יש לקלוט את הפרמטר שנשלח אלינו. אנחנו יודעים לקלוט פרמטר, ואפילו בשתי שיטות : GET ו- POST. השאלה : באיזו שיטה נשלח אלינו הפרמטר? לכאורה בעיה. הרי לא בנינו טופס בכלל, וממילא לא כתבנו שיטה מסוימת. רמז גדול נקבל אם נתבונן בשורת הכתובת של הדף bookdetail.asp כשהוא נקרא.

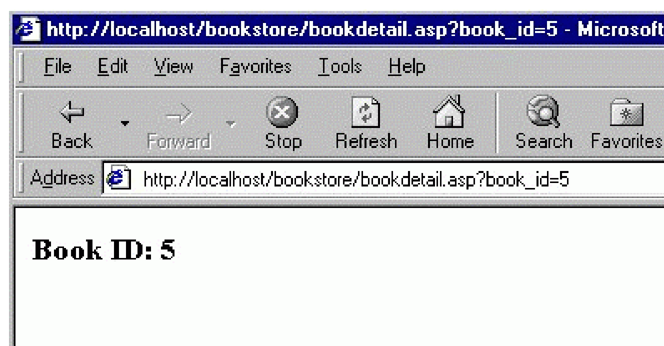
זה נראה כך:



הפרמטר משורשר בתוך מחרוזת הכתובת. וזהו, כזכור לכם, סימן ההיכר של שיטת GET. לכן, כדי לקבל את הפרמטר בתוך הקוד שלנו, נכתוב שורות דומות מאוד לדוגמה Get2.asp:

```
<%
dim book_id
book_id=Request.QueryString("book_id")
Response.Write "<h3>Book ID: " & book_id & "</h3>"
%>
```

נריץ ונבדוק. שימו לב, אי אפשר סתם להריץ את bookdetail.asp! ההרצה כאן אינה פשוטה, כי יש לשלוח פרמטר! נשמור את עבודתנו, נעבור לדף booksearch.asp ושם נלחץ לחיצה ימנית ונבחר View in browser. נבצע חיפוש לספר כלשהו, נקבל רשימה, נבחר ספר ונלחץ פרטים. נצפה כמובן לראות משהו הדומה לזה:



כעת, כשאנו בטוחים שהפרמטר שנקלט הוא הנכון, אפשר להשתמש בו. אנו רוצים להביא את כל הפרטים על הספר, כשנתון לנו קוד הספר. ובשפת SQL (לא להקליד):

```
select * from books where book_id=5
```

הבעיה שלא תמיד יהיה קוד הספר 5, לכן במקום לכתוב ישירות את המספר, נשרשר למחרוזת את המשתנה שקלטנו. השאילתה תיראה כך (עדיין לא להקליד):

```
sql="select * from books where book_id=" & book_id
```

כמו בכל הבאת מידע ממסד הנתונים, אנו זקוקים לקישור ורשימה. נבצע זאת בדיוק כפי שראינו בדוגמאות הקודמות. הקלידו את השורות הבאות:

```
dim conn
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"

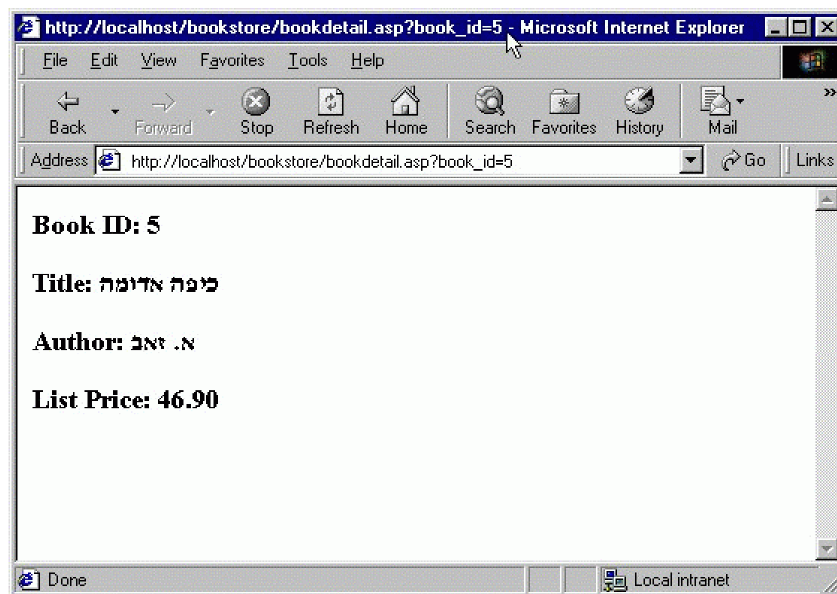
dim rs, sql
set rs=Server.CreateObject("ADODB.recordset")
sql="select * from books where book_id=" & book_id
rs.Open sql, conn
```

בדרך כלל, בשלב זה הינו מתחילים לסרוק את recordset בלולאה. אלא שכאן, אין צורך. ביקשנו שורה בודדת, וקיבלנו. rs מכילה שורה אחת בלבד של הספר המבוקש. לכן ניתן לכתוב בפשטות כך (הקלידו בהמשך לקוד הקודם):

```
Response.Write "<h3>Title: " & rs("title") & "</h3>"
Response.Write "<h3>Author: " & rs("author")& "</h3>"
Response.Write "<h3>List Price: " & rs("list_price")& "</h3>"
```

נריץ ונבדוק אם אכן מגיעים כל פרטי הספר. אני מזכיר שוב: לא להריץ ישירות את דף הפרטים (תבוא הודעת שגיאה על חוסר בפרמטר), אלא את דף החיפוש (searchbook.asp), ממנו להגיע לרשימה (booklist.asp), ובה ללחוץ על "פרטים נוספים".

הדף ייראה בערך כך:



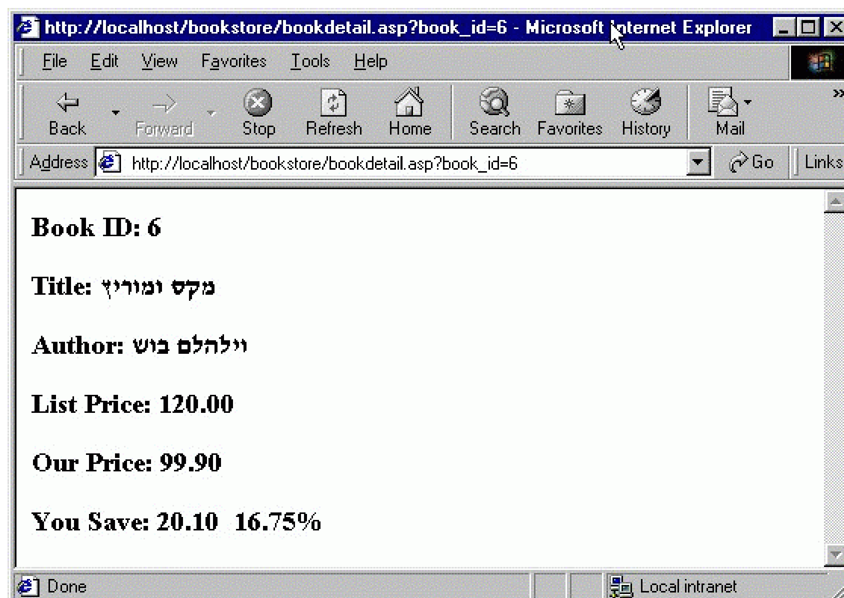
אני מסכים שמבחינה ויזואלית זהו דף מכוער ביותר, אבל מבחינת שליפת המידע, הוא מדויק. בעוד כמה דקות נעשה לדף הפרטים "מתיחת פנים" והוא ייראה קצת יותר מושך. בינתיים, עוד קצת מידע על הספר.

מקובל באתרי מכירות להציג את המחיר הרגיל (מחיר מופקע, בדרך כלל), המחיר שלנו (הנמוך יותר, מיותר לומר), ולהראות כמה אתה חוסך בקנייתך אצלנו.

גם לנו יש מחיר (price) ומחיר מחירון (list_price). את ההפרש נמצא (כן, זה מדהים) על ידי חיסור. מחקו את שורת המחיר הקיימת והקלידו בבקשה:

```
Response.Write "<h3>List Price: " & rs("list_price") & "</h3>"
Response.Write "<h3>Our Price: " & rs ("price") & "</h3>"
Response.Write "<h3>You Save: " & rs ("list_price") – rs("price") & "</h3>"
```

אתגר לאמיצים - חשבו והציגו גם את החיסכון באחוזים. הדף שלכם צריך להיראות כך:



כעת לתמונה. ניתן לשמור תמונות בכמה דרכים. דרך אחת היא לשמור את התמונה בתוך מסד הנתונים. זה אפשרי גם ב-Access, וסוג הנתונים המיועד לכך הוא OLE Object. אך יש גם דרך אחרת, פשוטה וחסכונית יותר. לפתוח תיקיה רגילה בשרת, ששמה יהיה, למשל, pictures, ובה יאוחסנו כל התמונות. אנו נשמור בטבלה רק שדה טקסט ובו שם התמונה. במקרים של הרבה תמונות, נרצה אולי לפצל את מאגר התמונות לתיקיות שונות, ואז נשמור לא רק את שם התמונה, אלא גם את כל המסלול (path) אליה.

לשיטה הראשונה, שמירת תמונות בתוך מסד הנתונים, יש יתרונות מסוימים:

1. קל לשלוף את התמונה, היא חלק מהנתונים.
2. גיבוי מסד הנתונים כולל גם את התמונות.
3. מעבר לשרת אחר, החלפת דיסק, או צרות חומרה אחרות - לא משפיעות.

אך יש לה גם חסרון כבד: התמונות תופסות נפח רב, והן תגדלנה באופן בלתי סביר את מסד הנתונים. כאשר מסד הנתונים יהיה גדול מאוד - זמן הגישה למידע יתארך אף הוא. לכן אנו מעדיפים בדרך כלל את השיטה השנייה. שומרים את התמונות כקבצים פשוטים במערכת ההפעלה, ובמסד הנתונים שומרים רק מחרוזת המכילה את שמם.

נעשה זאת כך. בטבלת books נוסיף, אם לא קיים כבר, שדה בשם picture שיהיה מסוג טקסט. בשדה זה נקליד את שמות תמונות הספרים. אם אין לכם תמונות כאלה, אתם יכולים להוריד מהאינטרנט (מאתרי ספרים, למשל מאתר הוד-עמי www.hod-ami.co.il), או להעתיק מהתקליטור המצורף לספר זה, כמה תמונות לצורך הלימוד, בתיקיה bookstore\Pictures.

פתחו תיקיה בדיסק שלכם, וקראו לה pictures. העבירו לתיקיה זו את תמונות הספרים שלכם. התיקיה חייבת להיות מתחת לפרויקט, וחייבת להיות "מוכרת" על ידי הפרויקט כדי שהקבצים יועתקו לשרת. מסלול התיקיה:

c:\inetpub\wwwroot\bookstore\pictures



שאלה: האם נשמור רק את שם הספר, או את כל המסלול?

תשובה: תלוי. בעיקרון ניתן לשמור רק את שם הספר, ולציין את המסלול פעם אחת בתוכנית. כך נעשה בתוכנית שלנו. באפליקציות אמיתיות עשויות להיות אלפי תמונות, ואז נרצה אולי לחלק אותן לתיקיות שונות: תיקיה לספרים, תיקיה לדיסקים, תיקיה לצעצועים וכו'. במקרים אלה נשמור גם את שם התיקיה, אבל לעולם לא מסלול מלא, מפני שלגולש שלנו אין תיקיה כזו. מסלול התמונות יהיה תמיד יחסי לפרויקט.

כך תיראה הטבלה שלכם, לאחר שהקלדתם את שמות הספרים:

picture	stock	price	list_price	author	title	book_id
alice1.jpg	100	45	100	לואיס קרול	עליסה בארץ הפלאות	1
snark.jpg	1000	12	14	לואיס קרול	ציד הסנארק	2
shirim_upizmonot.jpg	10	110	123	ח. נ. ביאליק	שירים ופזמונות לילדים	3
gan_gurim.jpg	100	8	10	רפאל ספורטה	גן גורים	4
kippa.jpg	120	129.9	149.9	א. זאב	כיפה אדומה	5
maxandmoriz.jpg	1	99.9	120	וילהלם בוש	מקס ומוריץ	6
alice_mirror.jpg	22	160	200	לואיס קרול	עליסה בארץ המראה	7
alice2.jpg	0	60	80	א. ב. יהושוע	הרפתקאות עליסה	8
mea_shanim.jpg	0	40	45	ל גרסיה מרקס	מאה שנים של בדידות	9
atlasshrugged.gif	0	48.9	70	ג'ון גאלט	מרד הנפילים	10

כאן חשוב לדייק! אם שם הספר בטבלה אינו זהה בדיוק לשם הקובץ, לא תראו את התמונה!

סגרו את הטבלה וחזרו לדף BookDetail.asp. הוסיפו לו, לאחר השורות האחרונות שכתבנו (מחירים), את השורה הבאה:

```
Response.Write "<IMG SRC='pictures\" & rs("picture") & "'>"
```

השורה מורכבת מ-3 חלקים (מוכר, אהה):

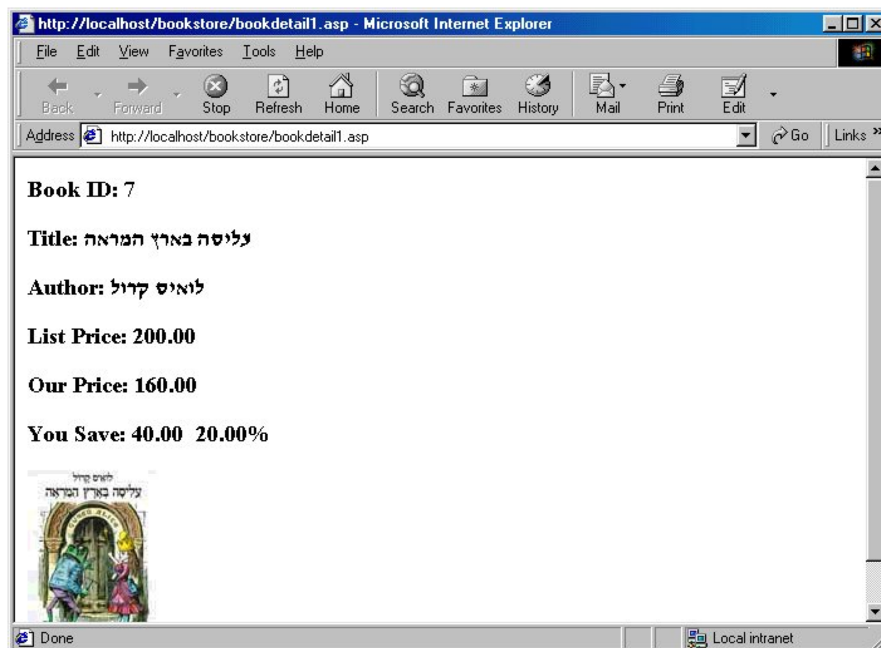
החלק הראשון כולל את תחילתה של התגית <img> ועם הנתיב של המאפיין src. שם התמונה הוא משתנה וזה יהיה החלק השני של המשפט, שכל תפקידו להביא את שם התמונה מהאובייקט rs. החלק השלישי מכיל רק את הסימן גרש (') והסימן >. הגרש סוגר את התוכן של המאפיין src והסימן > סוגר את התגית <img>.

מובן שאם בחרתם תיקיה אחרת לתמונות, כתבו אותה במקום \pictures. מה תהיה התוצאה? פקודת IMG רגילה של HTML. כשתריצו, תראו את התמונה של הספר הנבחר. הציצו בקוד המקור (View > Source) ותראו:

```
<IMG SRC='pictures\alice_mirror.jpg'>
```

שימו לב לגרשים שמקיפים את המסלול ושם התמונה. ודאו שכתבתם אותם בקוד המקור.

כך צריך להיראות הדף לאחר הוספת התמונה:



עיצוב דף הספר הבודד

יפה. כל הפרטים נמצאים על הדף. הבעיה היחידה היא, שהם לא מסודרים כפי שאנו רוצים. סידור הפרטים על פני המסך, הוא כמובן עניין של טעם אישי. לפני שמתחילים לעבוד, ננסה לצייר תרשים קטן:

<table>
<tr><td>שם הספר</td><td rowspan="5">תמונת
הספר</td></tr>
<tr><td>שם המחבר</td></tr>
<tr><td>מחיר רגיל</td></tr>
<tr><td>מחיר אצלנו</td></tr>
<tr><td>החסכון שלך</td></tr>
<tr><td colspan="2">תקציר הספר</td></tr>
</table>

הדרך הקלה ביותר להשגת תוצאת עימוד כזאת, היא באמצעות טבלאות HTML. כרגיל בטבלאות, יש ללכת מהגדול לקטן: ראשית לבנות את הטבלה החיצונית, טבלה בת שתי שורות. השורה התחתונה היא תקציר הספר. השורה העליונה תחולק לשתי עמודות (td). בשמאלית תהיה התמונה, ובימנית תהיה טבלה נוספת בת 5 שורות: שם הספר, המחבר, מחיר וכן הלאה.

למדנו לכתוב פקודות HTML מתוך ASP, אבל בואו נודה: זה הרבה יותר קל ב- HTML. לכן נעשה דבר פשוט: בחלק הראשון, נקרא את כל הנתונים הדרושים לנו מתוך מסד הנתונים לתוך משתנים. נסיים לקרוא, נסגור את הקישור, ונעבור ל- HTML. כאן נבנה את הטבלאות בקלות רבה. ובתוך הטבלאות נשתול את המשתנים האוגרים את נתוני הספר.

ולעבודה: בדף bookdetail.asp, מחקו את כל מה שהקלדנו עד כה בין <body> ל- </body>, והקלידו לתוכו מייד לאחר התג <body>, כך (קוד המקור מופיע ב- bookdetail2.asp):

```
<%
dim book_id
book_id=Request.QueryString("book_id")
```

```
dim conn
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"

dim rs, sql
set rs=Server.CreateObject("ADODB.recordset")
sql="select * from books where book_id=" & book_id
rs.Open sql, conn

dim title, author, price, list_price, review, you_save, picture

title = rs("title")
author= rs("author")
price = Formatcurrency(rs("price"),2)
list_price = Formatcurrency(rs("list_price"),2)
you_save = FormatPercent((rs("list_price")-rs("price")) / rs("list_price"), 2) &_
"    " & Formatcurrency((rs("list_price") - rs("price")),2)
review = rs("review")
picture = rs("picture")

rs.Close
set rs=nothing
conn.Close
set conn=nothing
%>
```

הסבר:

בשתי השורות הראשונות קולטים פרמטר. דף זה אמור להציג ספר בודד. חובה לתת לו את מספר הספר, אחרת הוא לא יידע את מי להציג (שאלה: באיזו שיטה נשלח הפרמטר, Post או Get ?).

לאחר מכן, פתיחה רגילה של קישור ורשימה. שימו לב לשאילתה: תן לי את כל פרטי הספר שהקוד שלו שווה לערך הפרמטר.



שאלה: כמה רשומות יוחזרו?

תשובה: רק אחת, כי רק ספר אחד יכול להיות בעל קוד מסוים.

לאחר מכן, מתחילה קריאה סדרתית של נתונים מתוך הרשומה שהתקבלה. שימו לב לפונקציה FormatCurrency, שמסדרת את הספרות במבנה המוכר של סכומים, ולפונקציה FormatPercent, שמציגה מספר בפורמט של אחוזים.



שאלה: כיצד אתה מעז להשתמש בפונקציות VBScript? אולי לגולשים שלי יש דפדפן netscape, שאינו מכיר את שפת VBScript ?

תשובה:

1. מה זה Netscape?!
2. הקוד שאנו כותבים ירוץ בשרת, לא במחשב הלקוח. השרת הוא שלנו ואנו קובעים באיזו שפה לכתוב. מה שיגיע לדפדפן הלקוח, יהיה HTML טהור. בדוק זאת. פונקציות אלו יבצעו את עבודתן רק בשרת.

נמשיך בעבודה, ונעבור לחלק השני.

כעת, ניצור טבלה ב- HTML. ניתן לעשות זאת ידנית, ויש גם דרך אוטומטית. הציבו את העכבר במקום בו אתם רוצים ליצור את הטבלה (לאחר סיום הקוד הקודם). בתפריט Table בחרו Table Insert. מופיע חלון:

קבעו את נתוני הטבלה :

רוחב : 550 פיקסלים.

מירכוז אופקי.

2 שורות.

2 עמודות.

עובי גבול 1.

לחצו על OK והנה זה פלא, נוצרה טבלה מקסימה.

הציבו את הסמן בשורה הראשונה, בעמודה השנייה, בין התגים <TD> ל </TD>, וצרו באותה דרך טבלה נוספת, עם הנתונים הבאים :

רוחב : 330 פיקסלים.

עובי קו : 1.

2 עמודות.

6 שורות (בחמש מהן נשתמש בפרק זה, אחת נשאיר לעתיד).

אשרו את החלון ובדקו את התוצאה. מדוע היינו צריכים בטבלה הפנימית שתי עמודות? אחת לכותרת ואחת לערך. לדוגמה, מחיר : 48 ש״ח.

כעת יש להקליד בתוך תאי הטבלאות את הערכים הנכונים. לא נשכח שאנחנו כעת בתוך ה-HTML, וכל אזכור של משתנה ASP צריך להיות בצורה <%=MyVar%>. לדוגמה, את תמונת הספר, נשבץ בתא השמאלי ביותר (הראשון) של השורה הראשונה. בין התגים <TD> ל- </TD> נכתוב כך :

```
<img src="c:\bookstore\pictures\<%=picture%>">
```



שאלה: מדוע ביקשתי עובי קו 1, הרי אנחנו לא רוצים להראות לגולש את קווי הגבול בין השדות?

תשובה: רק לנוחיות העבודה, כדי שנראה את התאים שלנו. אחרי שהכל יעבוד לשביעות רצוננו, נשנה את העובי לאפס, והקווים ייעלמו.

להלן קוד החלק השני. המשיכו להקליד בדף bookdetail.asp :

```
<center>
<table WIDTH="550" BORDER="1" CELLSPACING="1" CELLPADDING="1">
<tr>
<td align="center" valign="top">
<img src="c:\bookstore\pictures\<%=picture%>"></td>

 <td align="right" valign="top" width="390">
<table WIDTH="330" BORDER="1" CELLSPACING="1" CELLPADDING="1">
<tr>
        <td colspan="2" align="right"><strong><%=title%></strong></td>
</tr>
<tr>
        <td width="230" align="right"><%=author%></td>
        <td align="right">מחבר</td>
</tr>
<tr>
        <td align="right"><strong><%=list_price%></strong></td>
        <td align="right">מחיר רגיל</td>
</tr>
<tr>
        <td align="right"><strong><font color="blue"><%=price%>
        </font></strong></td>
        <td align="right">המחיר שלנו</td>
</tr>
<tr>
        <td align="right"><strong><font color="red"><%=you_save%>
        </font></strong></td>
        <td align="right">הרווח שלך</td>
</tr>

</table>
</td>

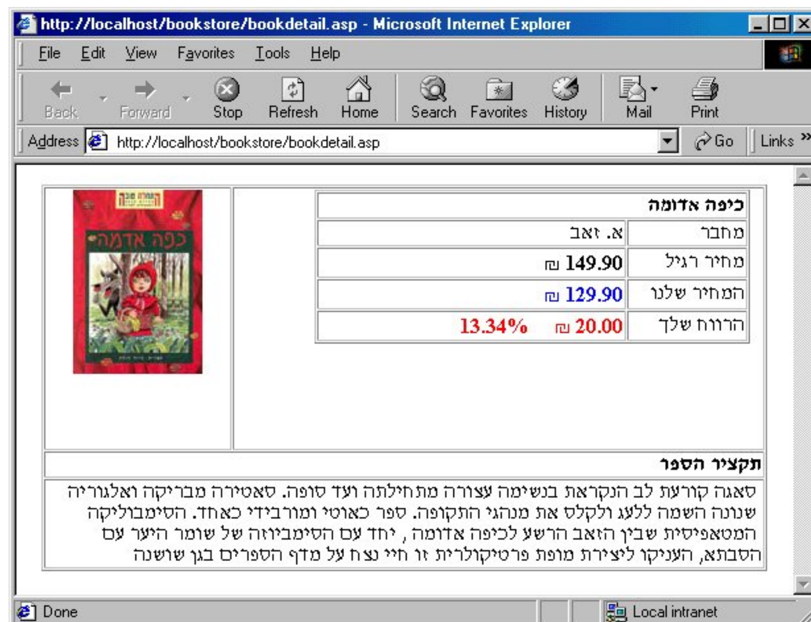
</tr>
<tr><td colspan="2" align="right"><strong>תקציר הספר</strong></td></tr>

<tr><td align="right" colspan="2"><%=review%></td></tr>
```

```
</table>
</center>

</body>
</html>
```

הדף צריך להיראות בסוף כך:



אם לא עובד לכם, בדקו:

1. שקראתם את המשתנים נכון. תמיד תוכלו לבדוק זאת בעזרת הדפסה ישירה למסך, כמו בגירסה הקודמת.
2. שלא שכחתם לכתוב שווה לפני הביטוי, בעת שיבוץ ב- HTML. דוגמה: <%=MyVar%>.
3. זכרו, הדף לא יכול לעבוד בפני עצמו. יש לקרוא לו מתוך רשימת תוצאות החיפוש.

אם לא מופיעה התמונה, בדקו את שם התמונה ואת המסלול המדויק. השוו בין שם התמונה בפועל לזה הכתוב בטבלה. נסו להדפיס את השם והמסלול יחד, וראו אם אין קו אלכסוני (/) מיותר או חסר. האם לא שכחתם את סיומת הקובץ? האם לא הפכתם בין JPG ל- GIF?

אם כן עובד לכם, זה הזמן ליהנות מפרי עמלכם, לנוח מעט, ולחשוב על כל ספרי "כיפה אדומה" שתמכרו באינטרנט.

פרק 7

זיהוי וספירת המשתמשים

הקובץ global.asa

עד עתה התקדמנו יפה ביישום האינטרנט שלנו. יש לנו כבר דפי ASP עובדים, אנו מבצעים שאילתות, מקבלים תשובות והכל טוב. אך חסר לנו "המימד האנושי". אנחנו לא מזהים את המשתמש הבודד ולא מסוגלים לטפל בו אישית.

כאשר בנינו את דף חיפוש הספרים ודף הרשימה עם הספרים המבוקשים, חשבנו על משתמש, שנניח מחפש ספרי בישול. מקליד מילה ומקבל רשימת ספרים. משתמש אחד, לא מאה משתמשים. אך מה יקרה לתוכנית שלנו, כאשר עשרה אנשים ייכנסו לדף החיפוש, וכל אחד יקיש מילה אחרת? כיצד יחליט השרת לאיזה מהם לענות? ומה יקבלו האחרים?

תחילה, נכיר בעובדות. ריבוי משתמשים הוא עובדת חיים ביישומי אינטרנט. ביישום חנות הספרים שלנו, עשויים להיות עשרה (או עשרת אלפים) אנשים בו-זמנית, שמחפשים ספרים, שניים שמזמינים כעת ומטפלים ב"עגלת הקניות" שלהם, חמישה שמזינים את פרטי כרטיס האשראי שלהם, ועוד חמישה-עשר שמעיינים במבצעים המיוחדים שלנו.

שרת IIS בנוי לכך ומטפל בכל משתמש כאילו הוא בן יחיד. עבור כל משתמש נפתח אובייקט הנקרא **Session**, שנפתח ברגע שהמשתמש נכנס לדף ASP כלשהו ששייך ליישום, ונסגר ברגע שהוא עוזב את היישום (עובר לאתר אחר, או סתם סוגר את הדפדפן). ה-**Session**, הוא האמצעי העיקרי בידי המתכנת ללוות את המשתמש ולשמור את מצבו: מה מספרו? באיזו שעה נכנס? איזה פריטים הוא כבר הזמין בעבר? האם שילם? מה יש בסל הקניות שלו כרגע? את כל התשובות לשאלות אלו, נאחסן במשתנים הנקראים Session Variables. מה שמייחד משתנים אלה הוא, שהם שייכים ל- **Session** מסוים, ולכן לגולש מסוים. הם תמיד זמינים לגולש לו הם שייכים, אך בלתי נגישים לכל גולש אחר.

אובייקט נוסף נקרא **Application**, והוא מרכז את כל המידע הנוגע ליישום כולו, ולא למשתמש בודד.

בקובץ מיוחד לכך, מרוכז הטיפול באובייקטים אלה. הקובץ נקרא **global.asa**, והוא נוצר אוטומטית כחלק מכל פרויקט Visual InterDev. חפשו אותו בפרויקט שלכם, ותמצאו אותו. הקובץ מטפל בארבעה אירועים. בינתיים נעסוק רק בשניים מהם:

Session_OnStart
Session_OnEnd

כלומר: התחלת וסוף **Session**. אנחנו נכתוב בקובץ global.asa שתי שגרות (פרוצדורות). האחת תרוץ עם תחילת ה- Session, כלומר כשהגולש התחבר אלינו, והשנייה עם סיום ה- Session, כשהוא נוטש אותנו. שגרה, היא בסך הכל חלק של תוכנית מחשב, שאפשר לקרוא לה כשרוצים, כלומר לגרום למחשב לבצע אותה. במקרה שלנו, המילה "כשרוצים" אינה מדויקת: השרת מפעיל שגרות אלו בדיוק כשצריך: האחת בתחילת ה- Session, והשנייה - בסופו.

המבנה העקרוני נראה כך:

```
Sub Session_OnStart

    ביצוע פעולות

End Sub
```

בתוך השגרה נכתוב פקודות שאנו רוצים שיתבצעו מייד עם הופעתו של הגולש החדש. מה נרצה? למשל הייתי רוצה לסמן את הגולש במספר מסוים. הייתי רוצה ליצור שורה בבסיס הנתונים ובה מצוינת העובדה שאדם נכנס לאתר, באיזה תאריך ובאיזו שעה. כשהוא ייצא, ארשום גם את שעת היציאה, וכך אוכל לדעת כמה גולשים היו אצלי, כמה זמן שהו בממוצע וכן הלאה.

כיצד יוצרים Session Variables? הנה, זה מאוד פשוט:

```
Session("VarName")=Value
```

דוגמה:

```
Session("user_name")="moshe"
```

אם קלטתי מהמשתמש את שמו, יהיה זה חכם מצדי לשמור אותו במשתנה Session, ואז בכל פעם שארצה אוכל לשלוף אותו ולהשתמש בו. כאשר אנו שומרים מידע באובייקט Session, הגולש מקבל אוטומטית מספר זיהוי ייחודי הנקרא **SessionID**. מספר זה מוקצה על ידי השרת והוא לקריאה בלבד, איננו יכולים לשנות אותו. מספר זה ילווה את הגולש בביקורו באתרנו. הוא שמור במשתנה Session, ואנו יכולים לקרוא אותו בכל עת, כך:

```
SID=Session.SessionID
```

לחילופין, ניתן גם ליצור עבור כל מבקר שורה בטבלה בה המפתח הראשי הוא מסוג Autonumber, מתקדם אוטומטית, ולהשתמש במספר זה כמספר מזהה לגולש.

דוגמה נוספת: ניצור משתנה, שישמור את תאריך ושעת הכניסה למערכת:

```
Session("EntryTime")=Now
```

כאמור, האובייקט Application מרכז את היישום, ונותן לנו אפשרות טיפול בשלושה נושאים:

1. מה לעשות עם תחילת היישום? ובעברית: השגרה Application_OnStart.
2. מה לעשות עם סיום היישום? ובעברית: השגרה Application _OnEnd.
3. ניהול משתני יישום. כאלה שאורך חייהם כאורך חיי היישום, והם נגישים לכולם.

אך ראשית יש להגדיר מהו היישום? מתי תחילתו, ומתי סופו? היישום הוא, כמובן, אוסף כל הדפים שכתבנו. "חייו" של היישום מתחילים ברגע שראשון הגולשים ביקש דף ASP מסוים השייך לו (רק דף ASP, דף HTML שכמובן יכול להיות לנו באתר, לא יגרום לאובייקט Application להיווצר!). גולשים נכנסים ויוצאים, והיישום ממשיך. היישום יסתיים רק כאשר הוא יסגר באופן יזום על ידי השרת.

השגרה Application_OnStart תתרחש ברגע תחילת קיום היישום, והיא מקום טוב לאתחל בו משתנים מסוג Application. בניגוד למשתני Session, שהם מיוחדים לגולש ספציפי, משתני יישום משותפים לכולם. יצירת משתנה Application:

```
Application("VarName")=Value
```

יצירת מונה מבקרים

מונה מבקרים הוא דוגמה קלאסית למשתנה יישום. בתוך Application_OnStart ניצור אותו, כך:

```
Sub Application_OnStart

Application("visitors")=0

End Sub
```

הקלידו את הקוד הנזכר למעלה לתוך הקובץ Global.asa בפרויקט שלכם.

בכל פעם שגולש חדש מגיע לאתר - אנו רוצים לקדם את המונה באחד. מהו המקום המתאים ביותר לכך? כמובן, השגרה Session_OnStart. בתוכה נשתול שורה כזאת:

```
Sub Session_OnStart

Application("visitors")=Application("visitors")+1

End Sub
```

בעברית: קדם את מונה המבקרים באחד. הקלידו גם שגרה זו לקובץ Global.asa שלכם. כל מבקר נוסף שייכנס לאתרנו, יקדם באחד את המונה שלנו.

מונה מסוג אחר שניתן לחשוב עליו, הוא מונה מבקרים בו-זמנית. לא כמה מבקרים היו אצלי מאז הקמת האתר, אלא כמה גולשים פעילים יש **כרגע**. אתחול המשתנה זהה לדוגמה הקודמת, וגם אותו יש לקדם עבור כל גולש חדש. השוני היחיד הוא שמשתנה זה יש גם להפחית באחד, בכל פעם שמישהו עוזב אותנו. להלן דוגמאות הקוד:

שלב 1, אתחול המשתנה:

```
Sub Application_OnStart
Application("active_visitors")=0
End Sub
```

שלב 2, קידום ב- 1 עבור כל גולש שמצטרף:

```
Sub Session_OnStart
Application("active_visitors")=Application("active_visitors")+1
End Sub
```

שלב 3, הפחתה ב- 1 עבור כל גולש נוטש:

```
Sub Session_OnEnd
Application("active_visitors")=Application("active_visitors")-1
End Sub
```

מה נעשה במשתנה זה? זה כבר תלוי בנו. אנו יכולים להראות אותו באתר לכל גולש, (אם יש לנו מה להראות...) ואנו יכולים לשמור אותו לעצמנו, למסכי אדמיניסטרציה בלבד. חשוב לדעת בכל פעם מה העומס על השרת. במקרים של נפילות או ירידה בביצועים, מעניין לדעת כמה גולשים היו, והיכן המגבלות שלנו.

את השורות המטפלות במשתנה ה-session הנקרא active_visitors, כפי שנראה בדוגמה, יש לשתול, כל אחת בתוך השגרה המתאימה לה. לסיכום, כך צריך להיראות הקובץ שלכם כשהוא כולל טיפול בשני המשתנים: מונה מבקרים כללי, ומונה מבקרים פעילים:

```
Sub Application_OnStart
Application("visitors")=0
Application("active_visitors")=0
End Sub

Sub Session_OnStart
Application("visitors")=Application("visitors")+1
Application("active_visitors")=Application("active_visitors")+1
End Sub

Sub Session_OnEnd
Application("active_visitors")=Application("active_visitors")-1
End Sub
```



שאלה: מדוע טיפלנו במשתנה visitors רק בפתיחת ה- session ולא בסיומו?

תשובה: מפני שאנו רוצים לדעת את מספר הגולשים המצטבר, לכן עלינו רק לקדם אותו עבור כל גולש חדש. עבור גולש שנוטש אותנו, אין לנו מה לעשות.

כעת אנו רוצים לראות את תוצאות עבודתנו. תחילה, הריצו את היישום כמו שהוא, רק כדי לוודא שאין הודעות שגיאה. אם הכל בסדר - נמשיך.



הערה:

את הקובץ global.asa אי אפשר להריץ, ואין לו שום חלק ויזואלי! כדי לבדוק שהקובץ תקין, הריצו כל דף אחר. אם יש לכם שגיאה כלשהי ב- global.asa, אתם כבר תשמעו על כך.

שגיאות נפוצות בכתיבת global.asa

1. ודאו שכל השגרות שלכם נמצאות בין <SCRIPT> ל- </SCRIPT>.
2. ודאו שאין לכם שתי שגרות באותו שם.
3. ודאו שכל שגרה מסתיימת ב- sub end. כדאי לשמור מרחק קטן בין השגרות כדי לראות טוב יותר.
4. בדרך כלל בכתיבת ASP אין חשיבות לאותיות קטנות או גדולות. באופן חריג, שמות השגרות צריכים להיות כתובים בדיוק כמו בספר, **עם הקפדה על אותיות קטנות וגדולות!**

צרו דף ASP חדש, לבדיקה בלבד, שייקרא למשל, test.asp. בדף זה כתבו:

```
Response.Write "You are visitor number: " & Application("visitors") & "<br>"
Response.Write "Active users now:  " & Application("active_visitors") & "<br>"
```

הריצו את הדף (לחיצה ימנית, view in browser). סביר שתראו שאתם מבקר מספר אחד, וכרגע יש מבקר אחד פעיל. זה הגיוני. סגרו את הדפדפן, ובצעו שנית. כעת תראו שאתם מבקר מספר שניים, ויש כרגע שני מבקרים. רגע, כיצד זה ייתכן? הרי ה"מבקר" הקודם כבר יצא! כנראה שהאירוע Session_OnEnd לא התבצע!

חשוב לדעת: Session מסתיים 20 דקות לאחר בקשתו האחרונה של המשתמש!

ה-Session של המשתמש הראשון, טרם הסתיים. חכו 20 דקות, הריצו שוב, ותראו שמספר המבקרים הפעילים ירד. השרת ראה שעברו 20 דקות ואין בקשה מהמשתמש הקודם, הפעיל עליו את Session_OnEnd, וכך ירד מספר המבקרים הפעילים.

ועוד חשוב לדעת: האובייקט Session נתמך רק על ידי דפדפנים התומכים ב-cookies.

ב-cookies נטפל בפרק הבא. בשלב זה יש לזכור שמנגנון ה-Session לא יפעל בדפדפנים ישנים מאוד, ולהביא זאת בחשבון.

פרק 8

מתכון לעוגיות שוקולד ללא אפיה

רוצים עוגיה? אני אוהב עוגיות חמאה עם שוקולד צ׳יפס. העוגיות שנדבר עליהן הפעם, יהיו טעימות פחות, אך שימושיות יותר. עוגיות - **Cookies**, הן קבצי טקסט קטנטנים, הנשלחים משרת האינטרנט למחשב הלקוח, ומתיישבים אצלו בדיסק הקשיח. אתם יכולים לעיין בדיסק שלכם, בעזרת סייר החלונות, בתיקיה c:\windows\cookies, ולראות עוגיות שנשלחו אליכם מאתרים שביקרתם בהם.

למה משמשות העוגיות? העוגיות משמשות בעיקר ככלי זיהוי בידינו, המתכנתים. לכל מבקר באתר שלנו אנו שולחים עוגיה משלנו, ובה פרטים מזהים אודותיו. כשייכנס אלינו שוב, נקרא ממנו את העוגיה, ונדע מיהו.

העוגיה משמשת לאחסון מידע במחשב הלקוח, וזו גם דרך מצוינת לשמור מידע ששייך ללקוח, או מאפיין אותו. לדוגמה, בפורטלים רבים יש לגולש אפשרות לאפיין לעצמו את חלון הכניסה. אילו נושאים מעניינים אותו, מה הסדר החביב עליו, באיזה צבע הוא רוצה את הרקע וכן הלאה. מידע זה יכול להיות מאוחסן בעוגיה (אחת או יותר), ועבור כל גולש שנכנס לאתר, תחילה, מחפשים את העוגיה. אם ישנה - קוראים ממנה את הנתונים ומציגים לגולש את מה שביקש. אם לא - זהו כנראה גולש חדש, ומראים לו את ברירת המחדל.

כיצד כותבים עוגיה למחשב הלקוח? זהו התחביר:

```
Response.Cookies("Cookie")=Value
```

דוגמה:

```
Response.Cookies("user_name")="Ronen"
```

וכיצד קוראים עוגיה ממחשב הלקוח אלינו?

```
var=Request.Cookies("Cookie")
```

דוגמה:

```
user_name=Request.Cookies("user_name")
```

עוגיה מורכבת

ניתן לשלוח עוגיה המורכבת מכמה יחידות מידע. להלן התחביר:

```
Response.Cookies("Cookie")("Key")=Value
```

דוגמה:

```
Response.Cookies("user_info")("fname")="Ronen"
Response.Cookies("user_info")("lname")="Almog"
Response.Cookies("user_info")("Account")="7114892"
```

הקריאה של עוגיה כזו דומה מאוד לקריאת עוגיה רגילה:

```
var=Request.Cookies("Cookie")("Key")
```

דוגמה:

```
fname = Request.Cookies("user_info")("fname")
lname = Request.Cookies("user_info")("lname")
account = Request.Cookies("user_info")("Account")
```



טיפ:
בכל כתיבה של עוגיה למחשב הלקוח, התסריט המבצע את הכתיבה חייב להיות קוד ASP הראשון, המבצע שליחה כלשהי למחשב הלקוח, הוא חייב להיות **לפני קוד HTML כלשהו!** . במילים אחרות לפני התגית <html>. ראו הסבר נוסף בפרק אודות איתור תקלות.

תאריך תפוגה

ידוע שעוגיות טעימות בעיקר כשהן טריות, וגם לעוגיות שלנו יש תאריך תפוגה. זהו מנגנון הגנה על המשתמש, המיועד לנקות את המחשב שלו מעוגיות ישנות. אם לא ניתן תאריך תפוגה לעוגיה ששלחנו, היא תימחק ממחשב הלקוח מייד עם עזיבתו את האתר! אם ברצוננו לשמור את העוגיה לביקור הבא של המשתמש, חובה לקבוע לה תאריך תפוגה רחוק יותר. לדוגמה:

```
Response.Cookies("user_name")="Ronen"
Response.Cookies("user_name").Expires = "01/04/2003"
```

ובעברית: קבע את תאריך התפוגה של העוגיה לתאריך קבוע וידוע מראש. בתאריך זה, תימחק העוגיה אוטומטית ממחשב הלקוח.

דוגמה נוספת:

```
Response.Cookies("user_name").Expires=Now() + 100
```

בדוגמה זו, קבענו את תאריך התפוגה לא לתאריך קבוע, אלא למאה ימים מיום המשלוח. יש לשיטה זו יתרון קטן על פני קביעת תאריך קבוע: אין צורך לתחזק את התוכנית ולעדכן את תאריכי התפוגה.

עשו לכם הרגל: מייד לאחר כל כתיבת עוגיה למחשב הלקוח, כתבו שורת תפוגה. כך תמנעו מעצמכם תקלות ואי הבנות.

מחיקת עוגיה

לעיתים רוצים למחוק עוגיה מסוימת ממחשב הלקוח. מעניין שאין לכך פקודה מיוחדת, אלא יש לבצע זאת בדרך עקיפה. אנו מנצלים את העובדה, שעוגיה נשארת בחיים רק אם תאריך התפוגה שלה מאוחר מהיום. כותבים תאריך תפוגה ישן, וזה הכל. מחשב הלקוח כבר יעשה את העבודה וימחק את העוגיה.

דוגמה:

```
Response.Cookies("user_name")=""
Response.Cookies("user_name").Expires = "01/01/1980"
```

והעוגיה נמחקה.

תרגול

צרו דף חדש, שייקרא, למשל, CookiesTest.asp. מחקו ממנו את כל מה שרשום, והקלידו את שורות הדוגמה שלנו:

```
<%
Response.Cookies("user_name")="Ronen"
Response.Cookies("user_name").Expires = Now() + 100
%>
```

כמובן שתוכלו לכתוב את שמכם במקום "Ronen".

הריצו. אם לא קיבלתם הודעות שגיאה כלשהן, והגיע דף ריק (הרי לא כתבנו כלום לחלון) אז כנראה הכל בסדר, ונוצרה עוגיה. בואו נחפש אותה. הפעילו את סייר החלונות, ובהנחה שאתם עובדים ב- Windows 9x, חפשו בתיקיה c:\windows\cookies. תזהו את העוגיה שלכם לפי התאריך והשעה בה נוצרה (סדרו לפי תאריך, ויהיה לכם קל יותר), וגם כמובן לפי שם השרת (במקרה שלנו שם השרת הוא bookstore). העוגיה היא, כאמור, קובץ טקסט. אתם יכולים לצפות בקובץ ולראות את תכולתו.



טיפ:

מן הסתם זו לא העוגיה היחידה במחשב שלכם. רוב האתרים הרציניים כיום שולחים עוגיות, וקשה יהיה לכם לאתר את העוגיה ששלחתם. כדי להקל על החיפוש, היכנסו בסייר החלונות לתפריט View (צפייה) ובחרו Details (פרטים) כלומר: לראות שורה מפורטת עבור כל קובץ. כעת לחצו על הכותרת Modified. הקבצים יסתדרו לפי תאריך יצירתם, והעוגיה שלכם תהיה ראשונה למעלה.

אתם יכולים לפתוח את קובץ העוגיה בעזרת NotePad ולראות את תכולתה.

כעת, כשווידאנו שאכן עוגיה נכתבה לדיסק שלנו, ננסה לקרוא אותה מתוך דף ASP.

תחילה, הפכו את השורות שכתבנו להערה. עושים זאת על ידי גרש (') לפני השורה. אם השורה הופכת לירוקה – הצלחתם. אם אינכם מצליחים, פשוט מחקו את השורות הקודמות. הקלידו את השורה הבאה:

```
Response.Write "My Name is: " & Request.Cookies("user_name ")
```

שמרו והריצו את הדף. אתם אמורים לראות משהו מעין זה:

```
My Name is: Ronen
```

אם הצלחתם לראות את השם ששלחתם בעוגיה – השגנו את מטרתנו. כתבנו מידע למחשב הלקוח, ושמרנו אותו בעוגיה. קראנו אותו כשרצינו בו. מצוידים ביכולת חשובה זו, נמשיך לשלב הבא.

פינת הגאון

שליחת עוגיה:

```
Response.Cookies("user_name")="Ronen"
Response.Cookies("user_name").Expires = Now() + 30
```

קריאת עוגיה:

```
user_name=Request.Cookies("user_name")
```

פרק 9

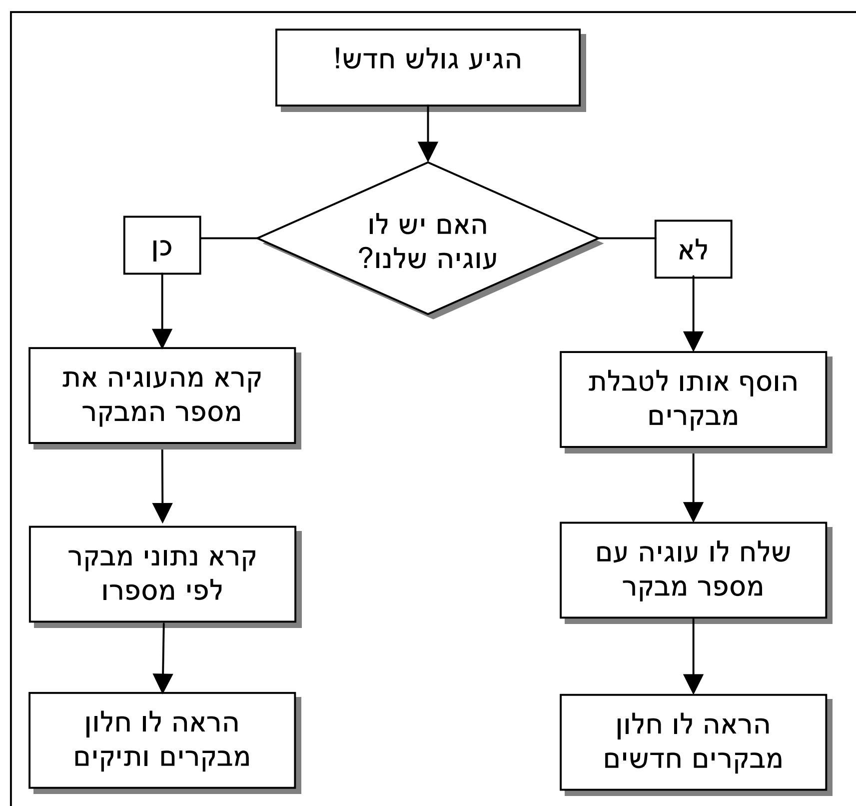
זיהוי מבקר חוזר

הרעיון הכללי

כמו כל בעל מכולת ממוצע, גם אנו רוצים להכיר את לקוחותינו. אנו רוצים להיות מסוגלים לזהות לקוח ותיק, לשמור את נתוניו, כך שלא ייאלץ להקלידם שוב ושוב. מצד אחד, אנו רוצים לנהל עבורו סל קניות שיוכל להישאר עבורו גם אם יצא מהאתר וחזר אחרי שבועיים. מצד שני, יש לנו מה להציע גם ללקוח חדש: הצעות להצטרף למועדון, מבצעי פיתוי מיוחדים שיגרמו לו להישאר ועוד.

באחד הפרקים הקודמים, הזכרתי את המשתנה SessionID. זהו מספר הניתן על ידי השרת לכל גולש המבקר באתר. לכאורה, טוב ויפה, בדיוק מה שאנחנו צריכים. מספר זהות לגולש. אלא שיש בעיה – בכל פעם שהגולש מגיע לאתרנו, הוא מקבל מספר חדש, לכן מספר זה יכול לשמש אותנו רק כל עוד האיש נמצא אצלנו. מרגע שעזב את האתר, אין עוד תועלת במספר זה.

לכן, הטכניקה שננקוט תהיה שונה במקצת, אם כי גם אנו נשתמש במשתנה SessionID. אנו ניצור טבלת לקוחות, ובה כל השדות הרגילים שצריכים להיות ללקוח: מספר לקוח, שם, E-mail, כתובת, מספר כרטיס אשראי, הכל. כאשר ייכנס גולש חדש, נוסיף אותו לרשימה. מספר המבקר שלו יהיה ה- SessionID שלו, וזה יהיה מספרו הקבוע. כדי שיהיה באמת קבוע, נשלח לו אותו בעוגיה. כעת, כאשר מגיע גולש לאתרנו, נבדוק: אם יש לו עוגיה שלנו עם מספר מבקר – זהו גולש ותיק, נשתמש במספר שלו. אם אין לו – זהו כנראה גולש חדש. נקצה לו מספר ונשלח לו עוגיה!



יצירת הטבלה

נתחיל בעבודה. למסד הנתונים שלנו (bookstore.mdb), נוסיף טבלה חדשה - טבלת visitors. להלן שמות השדות וסוגי הנתונים הדרושים:

Field Name	**Data Type**
visitor_id	Number Long Integer
entry_time	Date/Time
Exit_time	Date/Time
Visitor_name	Text 50
email	Text 50
address	Text 50
phone	Text 50

Field Name	Data Type
id	Text 50
credit_card_id	Number Long Integer
credit_card_num	Text 50
credit_card_exp_date	Date/Time



הערות:

1. מפתח ראשי: visitor_id.
2. ערך ברירת המחדל (default value) של שעת הכניסה יהיה ()now.

הסבר לתפקידי השדות

visitor_id – מספר מבקר. זה יהיה מספר הזיהוי של המבקרים באתרנו. זהו המפתח הראשי של הטבלה, אסור שיהיו בו כפילויות.

entry_time – תאריך ושעת כניסה. ננצל את תכונת ברירת המחדל, ונקבל רישום אוטומטי, ללא כל מאמץ מצידנו, של תאריך ושעת הכניסה. כל מה שצריך לעשות הוא לכתוב בתכונה default : ()now.

exit_time – תאריך ושעת נטישה. כאן לא יעזרו פטנטים, נאלץ לכתוב זאת בעצמנו.

E-mail – כתובת דואר אלקטרוני של הגולש, כדי שנוכל לשלוח לו דואר.

address, phone, id – כתובת, מספר טלפון, מספר זהות.

credit_card_id – סוג כרטיס אשראי. תהיה לנו טבלה קטנה עם סוגי הכרטיסים.

credit_card_num – מספר כרטיס אשראי. בחרתי בסוג טקסט ולא מספר, מפני שיש מספרי כרטיס ארוכים מאוד שלא ייכנסו ל- long integer.

credit_card_exp_date – תוקף כרטיס אשראי. חובה לציין אותו בעת מסירת דיווח לחברות האשראי. השימוש בשדה תאריך נעשה כאן רק לשם הדוגמה. בדרך כלל, חברות האשראי מסתפקות במחרוזת בת 4 תווים - mmyy. לדוגמה, 0402 פירושו חודש רביעי בשנת 2002.

כתיבת הקוד

כעת עלינו לכתוב את האלגוריתם שהצגנו בחלק א׳, אך היכן? היכן נגלה שהגיע גולש חדש? היכן נחליט אם הוא חדש או ותיק? המקום הטבעי לכתוב את כל זאת, הוא כמובן הקובץ Global.asa, בתוך השגרה Session_OnStart. כזכור לכם, פונקציה זו מופעלת פעם אחת עבור כל גולש, ברגע כניסתו. לא משנה לאיזה דף הוא נכנס. לפני שהוא מקבל את הדף הראשון, הוא חייב לעבור דרך השגרה Session_OnStart.

היכנסו לקובץ Global.asa וכתבו בו, אחרי כל שורות ההסבר המופיעות בצבע ירוק, כך:

```
Sub Session_OnStart

Application.Lock
Application("visitors")=Application("visitors")+1
Application("active_visitors")=Application("active_visitors")+1
Application.UnLock

session("visitor_id")=Request.Cookies("visitor_id")
```

ארבע השורות הראשונות מוכרות לכם. הן אלה שמקדמות את מונה המבקרים ומונה המבקרים הפעילים. לפניהן ניתן לראות את השורה application.lock ואחריהן application.unlock. יש לזכור שהאובייקט Application משותף לכל הגולשים. אנו רוצים למנוע מצב בו שני גולשים מעדכנים בו-זמנית את משתני האובייקט. לכן ״נועלים״ אותם לשבריר שנייה, מבצעים את הפעולה, ומשחררים את הנעילה.

השורה הבאה קוראת עוגיה ממחשב הלקוח, ומכניסה אותה למשתנה session("visitor_id"). על סמך משתנה זה נחליט אם זהו גולש חדש או ותיק. המשיכו להקליד:

```
dim conn
dim rs, sql
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"
```

זוהי פתיחת קישור סטנדרטית ביותר. נמשיך להקליד:

```
if session("visitor_id")="" then
	conn.Execute "insert into visitors (visitor_id) values(" & Session.SessionID & ")"
	session("visitor_id")= Session.SessionID
	Response.Cookies("visitor_id")= Session.SessionID
	Response.Cookies("visitor_id").expires = now() + 365
```

הסבר:

אם המשתנה session("visitor_id") ריק, זה אומר שלא הצלחנו לקרוא עוגיה מהלקוח. כלומר: זהו לקוח חדש. לכן אנחנו מכניסים שורה חדשה לטבלת visitors, בעזרת פקודת insert. כמספר מבקר אנחנו מכניסים את **Session.SessionID**, אותו מספר ייחודי הניתן על ידי השרת. אנו גם שומרים אותו במשתנה **Session("visitor_id")**, שילווה אותנו לכל אורך הדרך.

לסיום, אנו דואגים לעתיד: שומרים את מספר המבקר בעוגיה במחשבו של הגולש. כותבים עוגיה, ולא שוכחים לציין את זמן התפוגה שלה: שנה מהיום.

המשיכו להקליד (מה, התעייפתם?):

```
Else
        Set rs=Server.CreateObject("ADODB.recordset")
        sql="select * from visitors where visitor_id=" & Session("visitor_id")
        rs.Open sql, conn
        Session("visitor_name")=rs("visitor_name")
        Session("email")=rs("email")
        Session("address")=rs("address")
        Session("phone")=rs("phone")
        rs.Close
        set rs=nothing

End if
```

אם המחרוזת שהגיעה מהעוגיה אינה ריקה, מה המשמעות? שזהו גולש ותיק ומוכר! בפעולת קריאת העוגיה, כבר מילאנו את המשתנה Session("visitor_id") בערך הנכון! כל מה שיש לעשות כעת הוא רק לקרוא את פרטי הגולש מתוך מסד הנתונים שלנו, ולהכניסם לתוך משתני Session, כדי שנוכל להשתמש בהם בשעת הצורך. זה בדיוק מה שמתבצע בשורות המופיעות למעלה.

לסיום, לא נשכח לסגור ולמחוק מהזיכרון את אובייקטי הקישור והרשימה בהם השתמשנו.

```
conn.Close
set conn=nothing

End Sub
```

זהו. זה כל הטריק הגדול. אחזור עליו בקצרה: קראנו עוגיה ממחשב הלקוח. אם לא מצאנו עוגיה – הסקנו שהאיש פשוט מעולם לא היה אצלנו, וביצענו לו הליך קליטה. הוספנו עבורו שורה בטבלת המבקרים. בתור מספר מבקר הוא קיבל מספר חדש: את הערך של Session.SessionID.

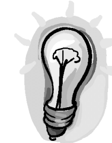
את המספר הזה שלחנו בתוך עוגיה, למחשבו של הלקוח. בפעם הבאה שיבקר אותנו, כבר לא יהיה חדש.

אם מצאנו עוגיה, סימן שהאיש כבר היה אצלנו בעבר. אין צורך להוסיף אותו לטבלת המבקרים. קראנו מתוך העוגיה את מספר המבקר, ולפי מספר זה קראנו גם פרטים נוספים על הלקוח.



שאלה: כל מה שאנחנו יודעים על המבקר בשלב זה, הוא מספר המבקר, מספר שקיבלנו מהשרת. כיצד אתה קורא את הטלפון, כתובת ושם? הרי מעולם לא הכנסתָ אותם למסד הנתונים?

תשובה: נכון. הנתונים שאקרא בשלב זה, יהיו ריקים. אבל אם יש לי מספר מבקר, אני יכול להציג לפני המבקר חלון הזנת פרטים, ולשמור אותם בשורה שלו בטבלה. אנו נעשה זאת בשלב ההזמנה (וגם בדף הצטרפות למועדון). כאשר האיש יאשר את ההזמנה ויבקש לשלם – זה יהיה הזמן לדרוש ממנו פרטים אישיים. כלומר, אם המבקר הוותיק קנה אצלנו פעם משהו, או הצטרף למועדון - אז הפרטים שלו קיימים אצלנו. אם הוא רק טייל באתר ולא קנה כלום – אלה יהיו שדות ריקים.



טיפ: הגנה פשוטה על דפים

לעיתים יש באתר דפים שהגישה אליהם מוגבלת. אנו רוצים להרשות גישה אליהם רק למי שעבר תהליך זיהוי כלשהו. התהליך שהודגם בפרק זה הוא אפשרות, אך ישנן גם אפשרויות אחרות. למשל, זיהוי על פי סיסמה, או מענה לשאלון כתנאי לכניסה לדף וכן הלאה. בדף אחד נקלוט את הסיסמה, בדף שני נבדוק אותה, ונחליט האם להפנות את הגולש לדף המידע, או לא. אבל, אם הגולש המתוחכם יודע את כתובת דף המידע שלכם, מי מונע ממנו להקליד ישירות את כתובת הדף ולהיכנס אליו? אנחנו נמנע ממנו. בתהליך הרישום נעדכן עבור הגולש משתנה session כלשהו. לדוגמה: ("session("username. כעת, בכניסה לכל דף ניתן לבדוק: האם יש ערך במשתנה זה? כלומר: האם הוא עבר תהליך קליטה נכון, או שזה אורח חסר זכויות?

```
<%
If session("username")="" then
  Response.End
End if
Response.write "Good Morning, " & session("username")
...
%>
```

פרק 10

הצטרפות למועדון הלקוחות

בפרק הקודם השגנו הישג חשוב – זיהינו את הגולש ונתנו לו מספר מזהה. כעת, כשיש בידינו כלי לזיהוי הגולש, ניתן לקלוט ממנו פרטים אישיים ולשמור אותם במאגר המידע. מועדון לקוחות, מועדון חברים, תחרות נושאת פרסים, הגרלה, משחק... כל אלה הם שמות יפים למטרה האמיתית: אנחנו רוצים את הפרטים שלך! למרות שהחוק מגביל את השימוש במאגר המידע, רוצים רוב מנהלי האתרים להגדיל ככל האפשר את "רשימת הלקוחות" שלהם, אפילו אם הם לא קנו דבר.

בדף שניצור כעת, נקלוט את פרטי הגולש, ונרשום אותם בטבלה. באיזו שורה נרשום אותם? לכל גולש באתר שלנו, לאחר שעבר בשלום את השגרה Session_OnStart, יש מספר מבקר, ויש שורה קיימת בטבלת Visitors. את השורה הזו נעדכן בפרטים שנקלוט ממנו.

כעת, לעבודה - צרו דף חדש וקראו לו joinclub.asp.

הציבו את הסמן מייד לאחר התג <BODY>, ובחרו בתפריט HTML > Form. קיבלתם שלד של Form ועליכם למלא את החסר. בתוך המרכאות של Action כתבו joinclub2.asp. כלומר, דף המטרה שלנו, הדף שאליו נשלח את הפרטים המוקלדים, יהיה joinclub2.asp.

פתחו הרבה מקום לפני סיום הטופס </form>, ובמקום זה כתבו את קלט הנתונים. כדי שהדף יהיה מסודר ויפה, ניצור טבלה. בתפריט Table בחרו Insert Table. יופיע חלון פרטי הטבלה. סמנו: 7 שורות, 2 עמודות, רוחב 300 פיקסל ומירכוז אופקי. בעמודה הימנית תהיה כותרת השדה, ובעמודה השמאלית – השדה עצמו.

בשורה הראשונה, לתוך העמודה השמאלית גררו רכיב Textbox, ושנו את שמו ל- visitor_name. בעמודה הימנית כתבו "שם מלא". הנה כך:

```
<TR>
	<TD align=right><INPUT id=text1 name=visitor_name></TD>
	<TD>שם מלא</TD>
</TR>
```

המשיכו כך עבור השדות: email, address, id, phone.

השדה הבא יהיה סוג הספרים שמעניינים את הגולש. אנו רוצים להציג בפניו תיבת בחירה. בעבר השתמשנו בתיבת בחירה, קומבו (combo box), אלא שאז הזנו את הנתונים ידנית. כעת נציג תיבת קומבו, שטוענת את ערכיה מתוך טבלה. לכאורה, לא בדיוק צריך את זה, כי קטגוריות הספרים אינן משתנות כל שבוע. אך מצד שני, כשהן משתנות, איננו רוצים לגעת בקוד. אנו רוצים שמנהל התפעול יוכל לשנות בעצמו קטגוריות, והוא ממש לא יודע ASP.

כעת, נעבור ל-Access, נפתח את מסד הנתונים שלנו, ונבצע בו שני שינויים:

1. נוסיף לו טבלה בשם categories, יהיו בה רק שני שדות:

   category_id – שדה מסוג מספר – קוד הקטגוריה (מפתח ראשי).

   category - שדה מסוג טקסט – שם הקטגוריה.

   נמלא בטבלה מספר ערכים, בדומה לתרשים:

categories : Table

category	category_id
ספרי מחשבים	1
ספרי בישול	2
ספרות יפה	3
ספרי ילדים	4
	0

2. נוסיף שדה לטבלת המבקרים שלנו - visitors, שבו נשמור את הקטגוריה המועדפת על הגולש. כדאי ונוח לתת לשדה את אותו שם, בדיוק כמו בטבלה אליו הוא מקושר – category_id. השדה יהיה כמובן מספרי, Long Integer.

נסגור את הטבלה ונחזור לדף שלנו. כולנו יודעים כיצד כותבים תיבת קומבו ב- HTML. ולמי ששכח, הנה תזכורת:

```
<SELECT id=select1 name=category_id>
<OPTION value=1>מחשבים</OPTION>
<OPTION value=2>ספרי בישול</OPTION>
<OPTION value=3>ספרות יפה</OPTION>
<OPTION value=4>ספרי ילדים</OPTION>
</SELECT>
```

(זו רק תזכורת כיצד צריך להיראות הקוד בסופו של דבר, לא להקליד!)

אם כך, נתרגם זאת ל- ASP: את השורות הראשונה והאחרונה נכתוב כלשונן, מילה במילה. ואת כל שורות האמצע, אלה עם ה- OPTION, נבנה דינמית מתוך נתוני הטבלה.

הקלידו בקובץ joinclub.asp, אחרי השורה החמישית בטבלה, אחרי התגית <TR>, את הקוד הבא:

```
<TR><TD align=right><SELECT id=select1 name=category_id
    style="HEIGHT: 22px; WIDTH: 150px">
<%
dim conn, sql, rs
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"
set rs=Server.CreateObject("ADODB.recordset")
sql="select * from categories order by category_id"
rs.Open sql, conn

Response.Write "<OPTION value=0>" & "בחר תחום" & "</OPTION>"
while not rs.EOF
  Response.Write "<OPTION value=" & rs("category_id") & ">" &_
                                                    rs("category") & "</OPTION>"
 rs.MoveNext
wend

rs.Close
Set rs=nothing
conn.close
Set conn=nothing
%>
   </SELECT></TD>
   <TD>תחום עניין</TD></TR>
```

הסבר:

1. את השורה הראשונה, הגדרת הקומבו, כתבנו עדיין ב- HTML. שימו לב להגדרות גודל התא.
2. עברנו ל-ASP, פתחנו קישור ורשימה. השאילתה מביאה את כל טבלת categories.
3. שורה אחת נכתבת עצמאית: קוד 0, והערך המוצג הוא "בחר תחום". זאת כדי, שהגולש יידע מה עליו לעשות. בדרך כלל משתמשים בטכניקה זו כאשר אין כותרת מלווה לקומבו.
4. עוברים בלולאה על כל הטבלה, ועבור כל שורה – כותבים שורת option לקומבו. הערך הוא rs("category_id"), והמלל המוצג הוא rs("category"). יש להבין שתיבת קומבו מכילה רשימה כפולה. רשימה של זוגות ערכים. ערך אחד לעיני המשתמש, וערך אחד לנו, המתכנתים. כאשר המשתמש רואה "ספרי בישול", אנו רואים 2. זהו הקוד של "ספרי בישול", והוא זה שיעבור הלאה לדף הקולט.
5. מחוץ ל- ASP, סוגרים את הגדרת הקומבו (Combo Box) וכותבים כותרת בעברית.

ניתן להריץ את הדף, ולוודא שהקומבו שלכם עובד יפה.

לא, עוד לא סיימנו. אני רוצה לדעת גם את המצב המשפחתי של הגולש. לשם כך נשתמש בלחצני רדיו. אלה נקראים כך כנראה לזכר אותם מכשירי רדיו ישנים, בדרך כלל במכוניות, שבהם לחיצה על לחצן אחד גרמה לשאר הלחצנים לקפוץ החוצה. כלומר, אי אפשר ללחוץ על שניים ביחד. זאת להבדיל מתיבות סימון (Check Box), שבהן ניתן לסמן כמה שרוצים.

דוגמה לשני הסוגים:

בצד ימין Radio Buttons

בצד שמאל Check Boxes

להביא את לחצני הרדיו לדף שלכם, זו משימה קלה ביותר, יש לגרור אותם מארגז הכלים לדף שלכם. רק ודאו שאתם אכן גוררים מארגז הכלים של HTML ולא של Design Time Controls. גם שם יש לחצנים ותיבות סימון (עם חבית צהובה קטנה), אבל אלה 'חיות' אחרות לגמרי.

גררו ארבעה לחצני רדיו מארגז הכלים, לשורה השישית והאחרונה בטבלה שלכם. הריצו את הדף, ותגלו שהלחצנים שלכם אינם מכירים זה את זה. ניתן ללחוץ על כל אחד מהם, ואת השאר זה לא בדיוק מעניין. כדי לשפר את הגיבוש ביניהם, תנו להם שם אחיד. כתבו לכולם, למשל: name=radio1. הריצו את הדף ובדקו, שאכן לחיצה על אחד משחררת את השאר.

כדי ליישר את הלחצנים ימינה, כמקובל בעברית, הוסיפו להגדרת העמודה את הקוד align=right. להלן הקוד המלא לארבעת הלחצנים:

```
<TR>
<TD align=right>
מחשבים וטכנולוגיה<INPUT type="radio" name=radio1 value=1 checked><BR>
חקלאות<INPUT type="radio" name=radio1 value=2 ><BR>
מסחר ושרותים<INPUT type="radio" name=radio1 value=3><BR>
עובד מדינה<INPUT type="radio" name=radio1 value=4><BR>
</TD>
<TD valign=top>תחום עיסוק</TD>
</TR>
```

שימו לב, לתוספת בלחצן הראשון, checked, הגורמת לו להיות מסומן כברירת מחדל.

זה כבר ממש הסוף. כעת נותר רק להוסיף לחצן Submit. ניתן לגרור לחצן כזה מארגז הכלים ולהסתפק בכך. אבל ניתן גם ליצור לחצן גרפי.



בתיקיה bookstore\images בתקליטור תמצאו לחצן בשם club.jpg. הוסיפו אותו לתיקיה images בפרויקט, והוסיפו אותו לפרויקט.

כעת הוסיפו שורה אחרונה לדף שלכם, לפני סוף ה- Form:

```
<input type="image" SRC="images/club.jpg" value="Submit" id="submit1"
name="submit1" WIDTH="190" HEIGHT="45">
```

הריצו את הדף שלכם, הוא צריך להיראות דומה לזה:

http://localhost/bookstore/personal1.asp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://localhost/bookstore/personal1.asp Go Links

שם מלא
Email
כתובת
מסי זהות
מסי טלפון
תחום עניין — בחר תחום
תחום עיסוק — מחשבים וטכנולוגיה / חקלאות / מסחר ושרותים / עובד מדינה

Done — Local intranet

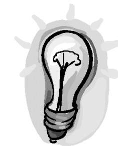
כרגיל, קווי הטבלה הושארו רק לצורך ההדגמה. כאשר אתם מרוצים מהעיצוב, הסירו את הקווים על ידי בחירת border=0 בטבלה.

יפה! כעת ניתן לבנות את הדף joinclub2.asp. זה הדף שיקבל את הפרטים שהוקלדו, ויכניס אותם למסד הנתונים.

צרו דף חדש שייקרא joinclub2.asp. מחקו את הכל, וכתבו כך:

```
<%
dim visitor_name, email, address, phone, credit_card_id
dim category_id, sql
visitor_name=Request.Form("visitor_name")
email=Request.Form("email")
address=Request.Form("address")
id=Request.Form("id")
phone=Request.Form("phone")
category_id=Request.Form("category_id")
radio1=Request.Form("radio1")
```

אלמנטרי, זכריה! המצאנו משתנים, וקלטנו אותם בזה אחר זה. תזכורת: הקליטה היא באמצעות Request.Form, מפני שהשיטה ב- form הקורא היתה Post!



טיפ:

השתדלו ככל האפשר לקרוא למשתנים בשמות הזהים לשמות השדות המתאימים בטבלה. אם שם הרכיב בטופס, שם המשתנה ושם השדה בטבלה יהיו זהים – הסיכויים שלכם לטעות קטנים בהרבה.

קלטנו את המשתנים, וכעת יש לבדוק אותם. בדוגמה זו, אנו בודקים רק אם השדות אינם ריקים. ניתן כמובן לבצע גם בדיקות אחרות, מורכבות יותר. כמו למשל: האם המידע אינו סותר? האם הוא חוקי? הקלידו:

```
if visitor_name = nil or email=nil or address=nil or phone=nil then
        Response.Write "<center><h2>"
        Response.Write "חלק מהפרטים שהקשת חסרים או שגויים"
        Response.Write "</h2></center>"
        Response.Write "<center><h2>"
        Response.Write "חזור לדף הקודם, תקן, ובצע שנית" & ""
        Response.Write "</h2></center>"
else
```

הסבר:

בדוק אם אחד המשתנים ריק (nil). אם כן – כתוב הודעת שגיאה, המסבירה למשתמש את טעותו. אפשרות נוספת: בדוק כל שדה בנפרד, ואז תוכל לכתוב הודעה מדויקת יותר, כמו: "שכחת להקליד שם משפחה. אנא חזור והקלד".

שימו לב למשפט התנאי: אם יש שדה לא חוקי – תן הודעה. אם לא (else), המשך הלאה. חובה לסגור ב- end if עם סיום הדף.

נמשיך להקליד:

```
dim conn
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"

sql="UPDATE visitors SET visitor_name='" & visitor_name
sql=sql & "', email='" & email
sql=sql & "', address='" & address
sql=sql & "', id='" & id
sql=sql & "', phone='" & phone
sql=sql & "', category_id=" & category_id
sql=sql & "   where visitor_id=" & session("visitor_id")

conn.Execute sql

conn.Close
set conn=nothing
%>
```



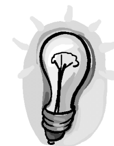
הערה:
לפני המילה where יש רווח. אם לא יהיה רווח תוצמד המילה where למספר שלפניה וזו תהיה שגיאת תחביר ב-SQL.

שלב ראשון: קישור.

לאחר מכן, בונים שאילתת עדכון. מדוע עדכון ולא הוספה? מפני שלכל גולש חדש, נוצרת שורה ברגע כניסתו לאתר. כל מה שצריך לעשות, הוא רק לעדכן את השדות הריקים בשורה זו.

השאילתה נבנית בשלבים. כך קל יותר לכתוב שאילתות ארוכות, ובעת הצורך גם למצוא בהן את השגיאה. השאילתה הסופית צריכה להיראות כך:

```
Update visitors SET visitor_name='dani', email='dani@shiri.co.il', address='herzel 77 bat
Yam', Id='44556677', phone='052-005522', category_id= 1
Where visitor_id=9876
```



טיפ:
עבור שדה המוגדר כ-String יש להקיף את הערך בגרשיים. עבור ערך מספרי – לא. חשוב להדגיש: מה שקובע אם שדה הוא מחרוזת או לא איננו דעתכם על אופי השדה, אלא הגדרתו בבסיס הנתונים!

לא תוכלו למצוא את השורה הזו בשום מקום. היא נוצרת ומתבצעת בשרת. במהלך הבנייה, משולב שם השדה במקום בו צריך להיות הערך. שימו לב לגרשיים! כל ערך שדה המוגדר במסד הנתונים כטקסט חייב להיות מוקף בגרשיים! עם סיום הכנת השאילתה, מפעילים אותה בשורה Conn.Execute sql. לא שוכחים לסגור את הקישור ולמחוק אותו מהזיכרון.

שימו לב, השאילתה מורכבת ובכתיבתה עשויות ליפול שגיאות כתיב. עצתי היא: הדפיסו את השאילתה. בין השורות הבונות את השאילתה, לבין זו המפעילה אותה, דחפו response.write sql. הקוד שלכם ייראה כך:

```
sql=sql & "   where visitor_id=" & session("visitor_id")
response.write sql
conn.Execute sql
```

ובמילים אחרות: טרחתם רבות לבנות שאילתת SQL. שנייה לפני שאתם מפעילים אותה, בואו נסתכל עליה. אולי חסר גרש לאחד המשתנים? אולי חסר רווח ואחת המילים השמורות צמודה למספר כלשהו? אולי חשבתם שאתם מקבלים פרמטר מדף קודם, אבל בעצם הוא לא הגיע, ויצא לכם =where visitor_id בלי שום דבר מצידו השני של השוויון? אם תעשו כך, רוב הסיכויים שתמצאו לבד את השגיאות.

לסיום, נודה לגולש על שהקדיש לנו מזמנו:

```
Response.Write "<center><h3>"
Response.Write "תודה על שהצטרפת למועדון"
Response.Write "</h3></center>"
```

ולא נשכח לסגור את ה- if:

```
end if
%>
```

רעיון נוסף לשיפור: אם הרישום למועדון הוא תנאי כניסה למסכים נוספים, כאן הוא המקום להפנות אליהם את הגולש. אם לדוגמה, רק חברי המועדון רשאים לראות את דף המבצעים, נוסיף כאן שורה: Response.Redirect "sales.asp", שתקפיץ אותו מייד לדף זה.

את עדכון הרשומה ניתן לבצע גם בשיטת ADO. מבחינת ביצועים כנראה שאין הבדל, וזו שיטה קלה יותר לכתיבה. פלוס גדול בעיני רבים הוא שאין התעסקות עם גרשיים. הנה הקוד לעדכון הרשומה, גרסת ADO:

```
sql="select * from visitors where visitor_id=" & session("visitor_id")
set rs=server.CreateObject("adodb.recordset")
rs.Open sql,conn,2,3

rs("visitor_name")=visitor_name
rs("email")=email
rs("address")=address
rs("id")=id
rs("phone")=phone
rs("category_id")=category_id

rs.update
rs.close
set rs=nothing
```

פינת הגאון: מילוי תיבת קומבו מתוך טבלה

שלב ראשון: בניית הרכיב (ניתן לגרור מארגז הכלים).

```
<SELECT id=select1 name=category_id
    style="HEIGHT: 22px; WIDTH: 150px">
```

שלב שני: הכנת ה- RecordSet.

יוצרים קישור ורשימה, בונים שאילתת SQL, ופותחים את הרשימה עם השאילתה.
חשוב מאד שהשאילתה תכיל שני שדות (לפחות): אחד לקוד והשני לערך המוצג.

```
<%
dim conn, sql, rs
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"
set rs=Server.CreateObject("ADODB.recordset")
sql="select * from categories order by category_id"
rs.Open sql, conn
```

שלב שלישי: סריקת הרשימה וכתיבת שורות Option.

```
while not rs.EOF
  Response.Write "<OPTION value=" & rs("category_id") & ">" &_
                                                    rs("category") & "</OPTION>"
  rs.MoveNext
wend
```

שלב רביעי: סיום. סגירת הרשימה והקישור, וסגירת רכיב הקלט.

```
rs.Close
Set rs=nothing
conn.close
Set conn=nothing
%>
</SELECT>
```



הערה:

הגדרת רכיב הקלט וסגירתו נעשו ב-HTML. יצירת השורות נעשתה ב- ASP.

יש כרטיס מועדון?

צירוף הפרק האחרון ופרק זה הביא אותנו לתוצאה מצוינת: מייד עם כניסת הלקוח לאתר – אנו מזהים אותו במספר לקוח. אם הוא נרשם במועדון (או קנה משהו) – יש לנו גם את שמו ופרטים נוספים.

אבל, מה קורה אם הוא עובר מחשב? אם הוא הולך לעבודה ונכנס לאתר שלנו ממקום העבודה? מה קורה אם הוא הולך לבקר את חמתו? (וגם זה דבר שקורה) - במחשבים אלה אין שום עוגיה שלנו. אנו נזהה אותו כלקוח חדש, ונשלח לו עוגיה חדשה. כל הפרטים שהוא טרח להקליד – הלכו לאיבוד. באתרים מסוימים זו לא בעיה רצינית, אך באתרים אחרים, בהם יש משמעות ל"ותיק" הלקוח, אנו רוצים לאפשר לו להיכנס כלקוח ספציפי במחשב חדש, ולהמשיך את רצף הפעילות כאילו הוא במחשב המקורי שבו עבד לראשונה. חשבו למשל על אתר בו ניתן לנהל תיק ניירות ערך, או אתר לחיפוש מקומות עבודה. אלה אתרים בהם הגולש משקיע זמן נכבד בהכנסת פרטים רבים, וכניסה לאתר כלקוח חדש עלולה לגרום לו תסכול רב. לכן אני מעוניין לתת בידי הגולש אפשרות להזדהות בכניסה לאתר.

מאוד פשוט: בדף הכניסה לאתר נשאל על שדה או שניים ונשווה את הנתונים לטבלת המבקרים שלנו. בדרך כלל נהוג לשאול שם וסיסמה. לנו אין סיסמה, ולכן נשאל שם וכתובת email.

צרו דף חדש, שייקרא index.asp. זה יכול להיות הדף הראשון באתר. העתיקו אליו את כל תוכנו של searchbook.asp. זו תהיה בעצם גירסה משופרת של searchbook הוותיק שלנו.

אני מזכיר לכם (דף או שניים אחורה) שבקובץ global.asa, אם גילינו שהמבקר ותיק, קראנו את פרטיו והכנסנו אותם למשתני session. הגיע הזמן להשתמש במידע זה. נכתוב את שם המבקר בדף הראשון, וכך הוא יידע אם זיהינו אותו או לא. אם אין שם כזה, נציע לו באדיבות להצטרף למועדוננו החביב.

לעבודה.

```
<div align=right>
<%
if session("visitor_name")<>"" then

' Old user
Response.Write "שלום " & session("visitor_name")
else
' New user
%>
שלום אורח, כדאי לך
<A HREF="joinclub.asp">להירשם</A>
לאתרנו
<br>
```

```
<%
end if
%>
```

עד כאן, די פשוט. אם יש לנו את שם המבקר, הצג אותו ואמור לו: "שלום חיים כהן". אם לא, אמור לו: "שלום אורח, כדאי לך להירשם לאתרנו." לחיצה על המילה "להירשם" תוביל אותו לדף joinclub.asp שכתבנו לא מזמן.

כעת, וחשוב להדגיש שזה קורה בכל אחד מהמקרים הנ"ל, אנו מציעים לו להקליד שם וכתובת email, ולגרום לזיהויו במערכת. כלומר, גם אם באתי לבקר חבר, וגם הוא גולש באתר זה, ובמחשב שלו כתוב "שלום אבנר לוי", אני עדיין יכול להקליד את פרטיי ולעבור ל"תיק" שלי.

נמשיך להקליד:

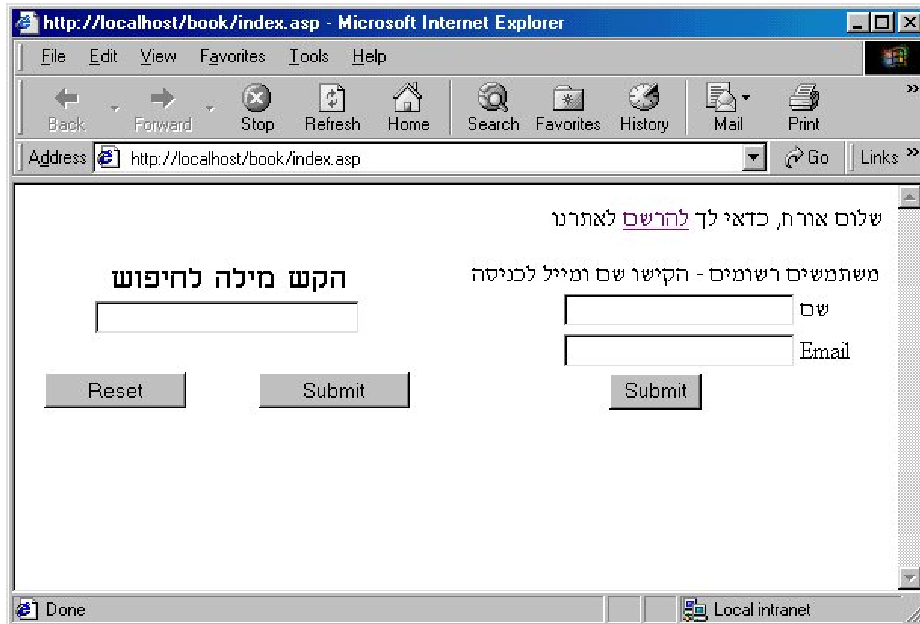
```
<FORM action="checkpass.asp" method=POST id=form1 name=form1>
<TABLE WIDTH=300
 ALIGN=right BORDER=0 CELLSPACING=1 CELLPADDING=1>
    <TR>
        <TD colspan=2 align="right">
        משתמשים רשומים - הקישו שם ומייל לכניסה</TD>
    </TR>
    <TR>
        <TD align="right">
        <INPUT type="text" id=text2 name="visitor_name"       dir="rtl"></TD>
        <TD>שם</TD>
    </TR>
    <TR>
        <TD align="right">
        <INPUT type="text" id=text3 name="email"></TD>
        <TD>Email</TD>
    </TR>
    <TR>
        <TD colspan=2 align=center>
        <INPUT type="submit" value="Submit" id=submit3 name=submit3>
        </TD>

    </TR>
</TABLE>
</FORM>

</div>
```

הסבר: זהו דף קליטת פרטים פשוט, בדיוק מהסוג שתרגלנו בתחילת הספר. יש בו שתי תיבות טקסט, ולחצן submit. שימו לב לתכונה action שמפנה את הפרטים המוקלדים לדף אחר, checkpass.asp.

הדף ייראה בערך כך בזמן ריצה:



ונעבור לדף הבדיקה. צרו דף חדש, וקראו לו checkpass.asp. בדף זה נבדוק אם השם וכתובת email שהקליד הגולש מופיע ברשימותינו. באותה קלות בדיוק אתם יכולים לבדוק סיסמאות.

הנה קוד הדף במלואו. הסברים אחריו.

```
<%@ Language=VBScript %>
<%
dim visitor_name, email
visitor_name=Request.Form("visitor_name")
email=Request.Form("email")
if visitor_name<>"" and email<>"" then

 dim conn
 set conn=Server.CreateObject("ADODB.connection")
 conn.open "dsn=bookstore"

 dim rs, sql
 set rs=Server.CreateObject("ADODB.recordset")
 sql="select * from visitors where visitor_name='" & visitor_name
 sql=sql & "' and email='" & email & "'"
 rs.Open sql, conn
 if rs.EOF then
   ' No good. Go back home, kid.
```

```
  Response.Redirect "index.asp"
 else
  ' OK, let him in.
 session("visitor_id")=rs("visitor_id")
 session("visitor_name")=rs("visitor_name")

 'Renew the Cookie
 Response.Cookies("visitor_id")=session("visitor_id")
 Response.Cookies("visitor_id").expires = now() + 30

 Response.Redirect "index.asp"

 end if

 rs.Close
 set rs=nothing
 conn.Close
 set conn=nothing

else
 Response.Redirect "index.asp"
end if
%>
```

הסברים:

1. ראשית – קלוט את שני הפרמטרים מהדף הקודם. אם לא נשלח כלום – חזור לדף הפתיחה.

(ניתן לשכלל ולתת הודעות שגיאה, אני משאיר את זה לכם...)

2. פתח קישור ורשימה, כמקובל, וצור שאילתה: שלוף את כל הלקוחות ששמם X וכתובת האינטרנט שלהם Y (שימו לב לגרשיים המקיפים את ערכי הטקסט!).

3. אם אין כאלה – חזור לדף הפתיחה.

4. אם יש כאלה (כמה יכולים להיות?) יש לעשות מספר פעולות:

א. קרא את שם המבקר ומספרו מתוך הטבלה, והכנס אותם למשתני session.

ב. שלח ללקוח עוגיה עם מספר המבקר הנ"ל.

ג. חזור לדף הפתיחה.

כעת נקליד את הפרטים בחלון,

http://localhost/book/index.asp - Microsoft Internet Explorer

שלום אורח, כדאי לך להרשם לאתרנו

משתמשים רשומים - הקישו שם ומייל לכניסה

שם רונן

Email ronen@netvision.net.il

Submit

הקש מילה לחיפוש

Submit Reset

והתוצאה תהיה זיהוי של המשתמש.

חזרנו לדף הראשי, אך הפעם יופיע שם המשתמש בראש הדף, וכך הוא יודע שהזיהוי הצליח.

http://localhost/book/index.asp - Microsoft Internet Explorer

שלום רונן

משתמשים רשומים - הקישו שם ומייל לכניסה

שם

Email

Submit

הקש מילה לחיפוש

Submit Reset

למרות שהזיהוי הצליח, תמיד יכול הגולש להקליד שם וכתובת אינטרנט אחרים, ולהיכנס כמשתמש אחר.

רעיון להמשך הפיתוח: בנו קישור נוסף: "שכחת את הסיסמה?" מי שילחץ על קישור זה, יוכל להקליד את כתובת האינטרנט שלו, ולקבל את סיסמתו בדואר אלקטרוני.

פרק 11

יצירת סל הקניות

הכנות לקראת סל הקניות

כל המאמצים שהשקענו בפרק הקודם, מכוונים למטרה אחת: לדעת עם מי אנחנו מדברים, או במילים אחרות, לקבל את המשתנה ("Session("visitor_id. כעת, כשמטרה זו הושגה, ניתן להשתמש בה. שימוש טיפוסי לזיהוי משתמש, הוא סל קניות. ניתן לממש סל קניות רק בסביבה בה המשתמש מזוהה בוודאות. אחרת, יהיה ערבוב בין קניותיהם של לקוחות שונים. הבסיס לסל קניות הוא טבלה פשוטה, נקרא לה basket, והיא תכיל את רשימת הקניות. במקומות אחרים תפגשו שמות כגון: Shopping Cart, עגלת קניות וכדומה. לאלה מכם שאינם מנהלי מסדי נתונים בעשר השנים האחרונות, הנה הסבר נוסף:

מבקר	ספר	הסבר
17	403	סל הקניות של מר לוי
17	539	
18	101	סל הקניות של גב' זליבנסקי
19	403	סל הקניות של גב' מועלם
19	175	
19	456	

יש לנהל טבלת קניות עבור כל גולש. אבל אין לנו כוח, זמן או חשק ליצור טבלה אמיתית לכל אחד. לכן ניצור טבלה אחת לכולם, כמו בטבלה למעלה. מר לוי, סימן שני ספרים: את מספר 403, ואת 539. לכן רשמנו בטבלה שתי שורות עבורו. לצורך זיהוי, נכתוב בכל אחת מהן את מספר המבקר של מר לוי, ואת מספר הספר. כשנרצה להראות למר לוי את תכולת סל הקניות שלו, נבצע סינון. נראה לו רק את השורות השייכות לו, אלה שמספר המבקר בהן הוא 17. בשפת SQL זה נשמע הרבה יותר טוב:

```
Select * from basket where visitor_id = 17
```

שימו לב לכך, שספר 403 מופיע ביותר מסל קניות אחד. הוא מופיע גם אצל מר לוי, וגם אצל גברת מועלם. אין בכך כמובן שום בעיה, זה כנראה ספר רב מכר ורבים רוצים בו. יש להקפיד בעת מחיקה מסל הקניות, למחוק את העותק הנכון: את זה ששייך למבקר שביקש את המחיקה.

וכעת לעבודה, במסד הנתונים שהקמנו (bookstore.mdb), הוסיפו טבלה חדשה בשם basket, עם השדות הבאים:

שם השדה	סוג נתונים	הסבר
basket_id	AutoNumber	מפתח ראשי. חסר חשיבות כשלעצמו.
visitor_id	Number, Long Integer	מספר מבקר
book_id	Number, Long Integer	מספר ספר
quantity	Number, Long Integer	כמות. ברירת מחדל: 1
orderdate	DateTime	תאריך רישום. ברירת מחדל: ()Now
status_id	Number, Long Integer	סטטוס השורה.ברירת מחדל: 0

שמרו את הטבלה בשם basket.

אל תשכחו לסמן את ברירת המחדל של התאריך בפונקציה ()Now, שפירושה תאריך של היום והשעה הנוכחית. בשדה כמות - ציינו את ברירת המחדל: 1. זוהי הכמות, אם לא ביקשו אחרת.

הסבר קצר על שדות הטבלה:

basket_id - זהו המפתח הראשי, אך אין לו כל שימוש כמספר. כדאי מאוד שיהיה מפתח ראשי, גם אם לא משתמשים בו ישירות.

visitor_id - זהו מספר המבקר. כאשר נרשום ספר בסל הקניות, נכתוב בעמודה זו את מספר המבקר הנוכחי. כאשר נבקש לקרוא את סל הקניות, נבקש רק שורות בהן visitor_id זהה למספר המבקר שלנו.

book_id - מספר הספר המבוקש. אין צורך לרשום את שם הספר, נביא אותו בשאילתה.

quantity - זו הכמות המבוקשת. כאמור, ברירת המחדל היא 1.

orderdate - תאריך הרישום (נמנעתי מלתת לשדה את השם date כי זו מילה שמורה, שימוש בה גורם צרות מאוחר יותר!). על סמך תאריך זה, ניתן יהיה לנקות את הטבלה מנודניקים, שסימנו ספרים ולא קנו אותם במשך שנתיים.

status_id - סטטוס השורה. חשבו לרגע: האיש סימן ספרים, והם נאספו בסל הקניות. הוא שילם. סל הקניות התרוקן. אך מה קרה לשורות שלו, האם הן נמחקו? חס וחלילה. אנחנו רוצים לעקוב אחר קניותיו של האיש, לעודד אותו לקנות עוד מסוג זה וכן הלאה. לכן ניתן סטטוס לשורה, ונחליט על ערכי סטטוס אפשריים.

לדוגמה:

0 - שורה חדשה, לא נקנה עדיין.

1 - נקנה, בטיפול מחלקת משלוחים.

2 - סופק.

על סמך נתון זה ניתן יהיה להפיק דוחות מעניינים, כמו:

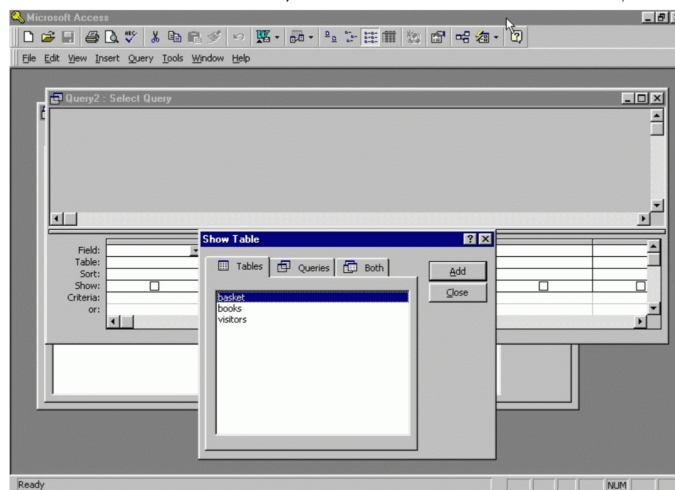
- דוח מכירות חודשי.
- רשימת ספרים בטיפול.
- רשימת הספרים שהוזמנו ושולמו לפני 72 שעות, וטרם סופקו.
- גרף מכירות לאורך שעות היממה, לאורך החודש, או לאורך השנה.

האמיצים שביניכם מוזמנים לנסות לייצר דוחות אלה. הזינו לטבלה שתיים או שלוש שורות לבדיקה. הזינו רק את השדות visitor_id ו- book_id והימנעו מלהזין נתונים לשאר השדות. בהנחה שיש לכם ספרים בעלי מספרים 1,2,3:

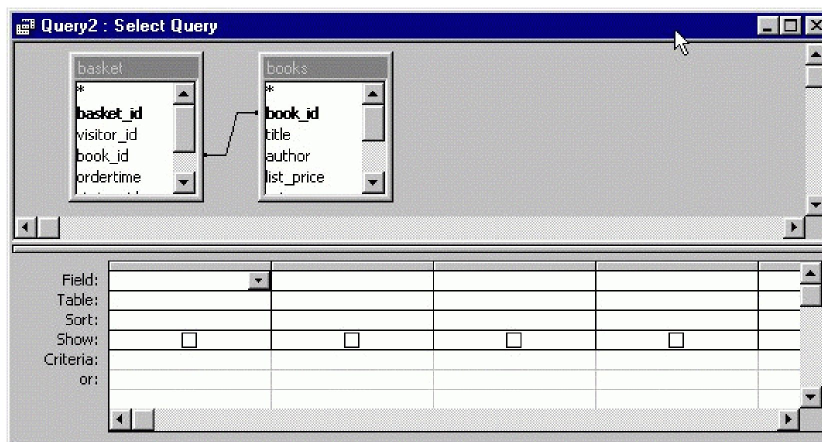
visitor_id	book_id
1	1
1	2
1	3

יצירת שאילתת סל הקניות

את הטבלה שזה עתה יצרנו, היינו רוצים מאוד להציג בדף ASP שיראה את כל מה שאסף הגולש עד כה. אך יש בעיה. הטבלה אינה מכילה את שם הספר, אלא רק את הקוד שלו. מי יתן לי את שם הספר? מי יכפיל עבורי את הכמות במחיר? שאילתה כמובן. נעבור לכרטיסיית השאילתות (Queries) ב-Access, נלחץ על לחצן New, ובחלון האפשרויות נבחר Design View.



קיבלנו חלון Select Query. אנו בוחרים שתי טבלאות: טבלת basket (אין צורך להסביר למה), ואת טבלת books, כדי לשלוף מתוכה את פרטי הספר: שם הספר, מחיר וכדומה. הוסיפו את שתי הטבלאות המופיעות למעלה בעזרת הלחצן Add, וסגרו את חלון הטבלאות. תמיד ניתן לקרוא לו שוב על ידי הסמל Show Table.
החלון צריך להיראות דומה לזה:

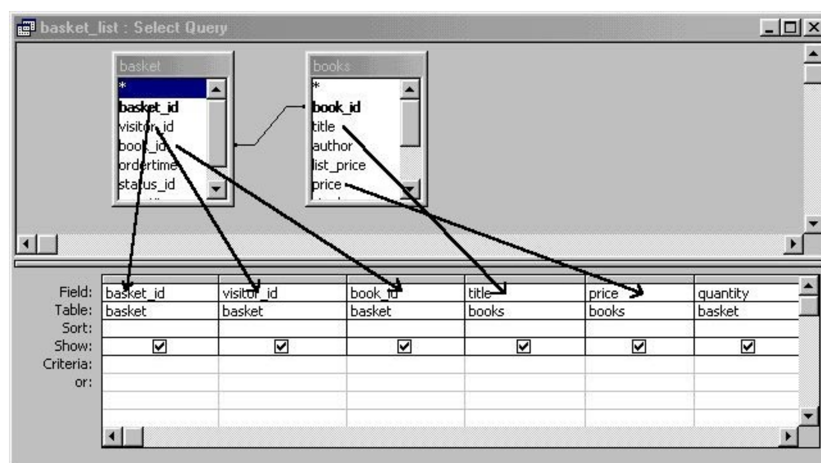


שימו לב לקישור ש-Access ביצע אוטומטית בין הטבלאות. הניחוש שלו הוא בהחלט נכון: אנו רוצים לשלוף נתונים מטבלת books על פי מספר הספר - book_id. אם לא מופיע אצלכם הקישור, בדקו ששמות השדות אצלכם נכונים, ושסוג הנתונים נכון.

כעת נתחיל לגרור שדות מתוך הטבלאות ל"גריד".

מטבלת basket נגרור את: quantity, book_id, visitor_id, basket_id.

מטבלת books נגרור את price, title.



לאחר שגררנו את השדות, ניתן לצפות בתוצאת השאילתה. לחצו על הסמל השמאלי ביותר בסרגל הכלים - Datasheet View (או לחיצה ימנית על כותרת השאילתה...), ותקבלו חלון דומה לזה:

Query2 : Select Query

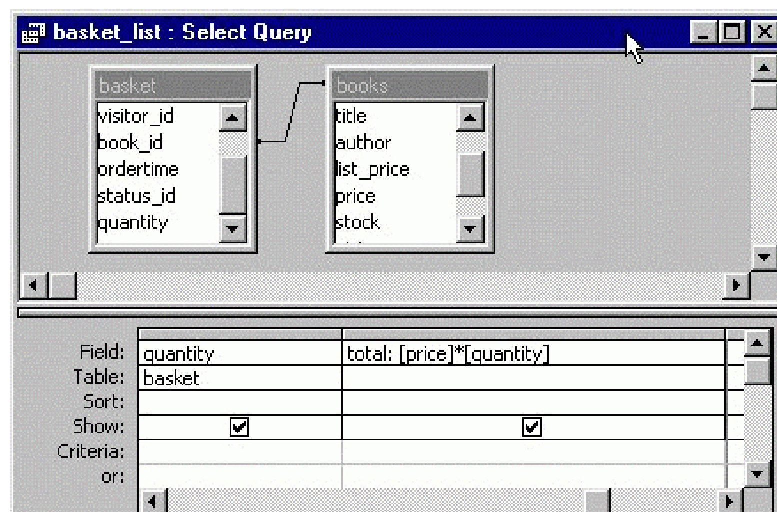
quantity	price	title	book_id	visitor_id
2	45	עליסה בארץ הפלאות	1	1
1	45	עליסה בארץ הפלאות	1	65
1	45	עליסה בארץ הפלאות	1	65
2	12	ציד הסנארק	2	2
1	12	ציד הסנארק	2	65
1	39.9	כיפה אדומה	5	66
1	40	מאה שנים של בדידות	9	65
1	40	מאה שנים של בדידות	9	66
1	18	ברנשים וחתיכות	11	65

Record: 1 of 9

מצוין! עבור כל ספר מסל הקניות (טבלת basket), קיבלנו את שמו ומחירו. חסר רק עוד דבר קטן אחד: מכפלת המחיר בכמות. ניתן לבצע זאת גם בדף ASP, אבל חשוב להכיר את שתי הדרכים.

בעמודה חדשה נקליד: **total: price * quantity**

והמשמעות: צור שדה מחושב, ששמו total, ושערכו יהיה מחיר כפול כמות.



צפו בתוצאה:

basket_list : Select Query

total	quantity	price	title	book_id	visitor_id
90	2	45	עליסה בארץ הפלאות	1	1
45	1	45	עליסה בארץ הפלאות	1	65
45	1	45	עליסה בארץ הפלאות	1	65
24	2	12	ציד הסנארק	2	2
12	1	12	ציד הסנארק	2	65
153	3	51	כיפה אדומה	5	66
40	1	40	מאה שנים של בדידות	9	65
40	1	40	מאה שנים של בדידות	9	66
18	1	18	ברנשים וחתיכות	11	65

Record: 1 of 9

ודאו שאכן תוצאת השדה total, היא המחיר כפול הכמות. ניתן לכתוב על תוצאת השאילתה. שנו את עמודת הכמות וראו כיצד משתנה הסכום.

שמרו את השאילתה בשם basket_list.

תזכורת: מותר להשאיר את Access פתוח ולעבוד במקביל, רק לא להשאיר את השאילתה במצב Design, מפני שאז היא נעולה ואי אפשר לעבוד איתה.

תוספת אחרונה: מה קורה עם ספרים שכבר נקנו? האם אנו רוצים להראות למשתמש בסל הקניות שלו ספרים, שהוא כבר קנה בעבר? מובן שלא. לכן יש להוסיף לשאילתה התניה: הראה רק ספרים שהסטטוס שלהם אפס. בזמן רכישה, נשנה את סטטוס הספר הנקנה ל-1, וכך יישאר סל הקניות שלנו מעודכן.

גרור מטבלת basket את השדה status_id, ובשורת הקריטריון נכתוב 0, כמו במסך שלפנינו:

basket_list : Select Query

Field:	visitor_id	book_id	title	price	quantity	total: [price]*[quantity	status_id
Table:	basket	basket	books	books	basket		basket
Sort:							
Show:	☑	☑	☑	☑	☑	☑	☐
Criteria:							0
or:							

כעת, שהשאילתה מוכנה, ניתן לחזור ל- Visual InterDev, ולהכין את דף סל הקניות.

הוספת סל הקניות

הוסיפו דף ASP חדש לפרויקט שלנו (שכחתם כיצד? לחיצה ימנית על שם הפרויקט, Active Server Page < Add), תנו לו שם: showbasket.asp. בדף שנוצר עבורכם ישנם תגי HTML - מחקו את כולם. מדוע למחוק? מפני שאנחנו עומדים להשתמש בפקודה Response.Redirect, וזוהי פקודה מאוד מפונקת, שאינה סובלת פקודות HTML לפניה. לאחר שמחקתם הכל, הקלידו (קוד המקור לדף זה נמצא בקובץ showbasket2.asp):

```
<%
dim conn
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"

dim rs, sql
set rs=Server.CreateObject("ADODB.recordset")
sql="select * from basket_list where visitor_id=" & session("visitor_id")

rs.Open sql, conn, 3
```

את שלוש השורות הראשונות, אין צורך להסביר, זוהי יצירת קישור סטנדרטית.

בפתיחת הרשימה שימו לב לבניית השאילתה ולהצמדת המשתנה ("session("visitor_id, כלומר: טבלת סל הקניות מכילה בעצם אלפי "סלים" של גולשים שונים, ואנו מציגים לגולש האחד שלנו רק את השורות **שלו**.

דבר נוסף שמחייב הסבר, הוא הספרה 3 שבסוף השורה האחרונה. לשיטה Open ישנם עוד כמה פרמטרים, ולא רק sql ו- conection, שהשתמשנו בהם. כאשר לא מציינים את ערכי פרמטרים אלה, נלקחים ערכי ברירת המחדל. הפרמטר השלישי הוא שיטת פתיחת הרשימה. ישנן כמה שיטות, וההבדלים ביניהן טמונים ביחס החליפין (trade off), שבין צריכת משאבים לבין יכולות. לדוגמה: השיטה בה נפתחת הרשימה בברירת המחדל, היא שיטה הנקראת OpenForwardOnly. זוהי שיטה חסכונית מאוד במשאבי השרת. כלומר, גולשים רבים יכולים לקבל דפים כאלה בלי להעמיס את השרת. בתמורה, כצפוי, אנו מוגבלים בפעולות שניתן לבצע על רשימה כזו. למשל: ניתן "לרוץ" על פני הרשימה קדימה בלבד. אין אפשרות לחזור אחורה. אין אפשרות לקבל את מספר הרשומות שחזרו בעזרת התכונה recordcount. אין אפשרות לעדכן את נתוני הרשימה. מגבלות כבדות. השיטה שבה אנו משתמשים בדף זה היא OpenStatic. שיטה ש"עולה" יותר במונחי משאבים, אך מאפשרת ריצה קדימה ואחורה, ועל ידי כך מאפשרת קבלת מספר הרשומות - recordcount.

להלן טבלת השיטות לפתיחת Recordset :

שיטה	קוד	הסבר
ForwardOnly	0	ברירת המחדל. חסכונית ביותר במשאבים. אינה מאפשרת תנועה אחורה ברשימה (rs.MovePrevious). אינה מתעדכנת בשינויים שביצעו משתמשים אחרים לאחר פתיחתה. אינה מאפשרת עדכונים.
Keyset	1	רשימה ניתנת לעדכון, המאפשרת תנועה בשני הכיוונים, קדימה ואחורה, אך איננה מתעדכנת בשינויי משתמשים אחרים.
Dynamic	2	מאפשרת תנועה על פני הרשימה בשני הכיוונים. מאפשרת עדכון הרשומות. משקפת שינויים שבוצעו על ידי משתמשים אחרים לאחר הפתיחה. זוהי שיטת הפתיחה הבזבזנית ביותר במשאבים. אינה נתמכת על ידי Access.
Static	3	מספקת עותק בלתי ניתן לעדכון של המידע. מאפשרת תנועה על פני הרשומות בשני הכיוונים. אינה משקפת שינויי משתמשים אחרים. שימושית בעיקר לשליפת מידע.

במקרה זה, כאמור, בחרנו בשיטה 3 - Static.

נמשיך להקליד :

```
if rs.EOF =true and rs.BOF=true then ' no book found
  Response.Redirect "basketisempty.asp"
end if
```

הסבר:

EOF - End Of File, היא תכונה המסמלת את סוף הרשימה. נתקלנו כבר ב- EOF כשיצרנו טבלה. לאחר כל שורה בדקנו אם כבר הגענו ל- EOF או לא. BOF הוא הקצה השני - Begining Of File. קיומם של שני אלה יחד, מעיד שזוהי רשימה ריקה! השורה שהקלדנו...If rs.EOF=true and rs.BOF then, היא שיטה מהירה וטובה לאבחן אם הרשימה ריקה.

ואם הרשימה אכן ריקה, מה זה אומר, ומה עלינו לעשות? רשימה ריקה, במקרה שלנו, פירושה סל קניות ריק. הגולש הנוכחי פשוט לא הזמין עדיין כלום. במקרה כזה נפנה אותו לדף מיוחד לכך. בדף זה ניתן לכתוב למשל: "סל הקניות שלך ריק....", להציג לו אפשרויות לחפש ספר על פי מילה, לגשת לדף המבצעים המיוחדים, או כל מה שאתם רוצים.

את הדף basketisempty.asp תכתבו בעצמכם, בזמנכם הפנוי.

גם הפקודה Response.Redirect שווה הסבר מיוחד. לכאורה פקודה פשוטה ביותר, השולחת את הגולש לדף אחר. הבעיה היא שהשליחה מתבצעת בעזרת חלק הכותרת - header - של פרוטוקול http. וכותרת כזו יש, מה לעשות, רק אחת לכל דף. אם נשלחה

כבר הכותרת - הפעולה תיכשל. הזכרתי כבר, בפרק הפתיחה, את אופן הטיפול של שרת האינטרנט בדף ASP : אם הוא נתקל בתגי HTML - שולח אותם מייד (עם header, אי אפשר בלי!). אם הוא נתקל בפקודות ASP, הוא מתחיל לבצע אותן בזו אחר זו. זוהי הסיבה שביקשתי למחוק את תגי HTML הראשונים מדף זה. הם היו גורמים לשליחת header ובכך מונעים את פקודת Redirect מלפעול! (ראה הסבר נוסף בפרק "איתור תקלות"). נמשיך להקליד :

```
bookcount=rs.RecordCount
Response.Write "<center><h3>" & "בסל הקניות שלך" & "  "
Response.Write bookcount & "  " & "ספרים" & "</h3>"
Response.Write "</center>"
```

בשורה הראשונה אנו מבררים כמה רשומות קיימות ב- rs, את התוצאה שומרים במשתנה bookcount. אז ניתן לכתוב את הכותרת : "בסל הקניות שלך 3 ספרים". במקום מספר הספרים יוצב כמובן המשתנה bookcount, שבמקרה הכנו מראש.

נמשיך להקליד :

```
Response.Write "<center><table border=1 cellpadding=2 BGCOLOR=Yellow>"
Response.Write "<tr>"
Response.Write "<td align=right width=50>מחיקה</td>"
Response.Write "<td align=right width=80>סך הכל</td>"
Response.Write "<td align=right width=50>כמות</td>"
Response.Write "<td align=right width=70>מחיר</td>"
Response.Write "<td align=right width=150>שם הספר</td>"
Response.Write "</tr>"
```

בשורות אלו יצרנו טבלה, וכתבנו את שורת הכותרות. לא סגרנו ב- </table>, מפני שיש לנו עוד שורות לכתוב. בסיומן נסגור את הטבלה. נמשיך :

```
do while not rs.EOF
 Response.Write "<tr>"

 Response.Write "<td  align=right>"
 Response.Write "<A HREF=" & "delfrombasket.asp?book_id=" & rs("book_id") & ">"
 Response.Write "מחק" & "</A></td>"

 Response.Write "<td align=right>" & formatcurrency(rs("total")) & "</td>"
 Response.Write "<td align=right>" & rs("quantity") & "</td>"
 Response.Write "<td align=right>" & formatcurrency(rs("price")) & "</td>"

 Response.Write "<td align=right><A HREF=" & "bookdetail.asp?book_id="
 Response.Write rs("book_id") & ">"
 Response.Write rs("title") & "</A></td>"
   Response.Write "</tr>"
 rs.MoveNext
loop
```

הסבר: כרגיל, בצע בלולאה עד סוף הרשימה:

צור עמודת מחיקה בסל הקניות. זהו קישור לדף מחיקה שנקרא delfrombasket.asp (אותו נבנה בהמשך). אנו כמובן קוראים לו עם פרמטר, שיידע איזה ספר למחוק. הפרמטר הוא, כמה מפתיע, ("rs("book_id. הכיתוב הוא, בפשטות, "מחק". המהדרין ישימו תמונה קטנה של איקס אדום, או של עיפרון עם מחק...



שאלה: מדוע אינני שולח את מספר המבקר? הרי אולי עוד אנשים הזמינו את אותו ספר? מה, נמחק אותו לכולם?

תשובה: חלילה. **אין צורך לשלוח את מספר המבקר, כי הוא קיים תמיד במשתנה ("Session("visitor_id**. בדף המחיקה (delfrombasket.asp), נמחק את כל השורות בהן מספר הספר שווה ל- book_id, **ומספר המבקר שווה ל- ("Session("visitor_id**.

עם זאת, שימו לב לנקודה מעניינת: אם המבקר סימן פעמיים את אותו ספר (דבר שאנו מרשים, אך ביישום האמת שלכם, אתם תחליטו), וביקש למחוק אחד מהם, שניהם יימחקו! מפני ששניהם עונים לקריטריון: מספר הספר, ומספר המבקר.

העמודה הבאה היא של הסכום הכולל. שימו לב, אנו מתייחסים אליו כאל שדה רגיל לחלוטין, למרות שהוא, כזכור לנו, שדה מחושב. שימו לב, גם לפונקציה שעוטפת את מקור המידע: FormatCurrency. היא עושה שני דברים: האחד, היא מסדרת את הספרות בצורה המקובלת בענייני כספים. לא סתם 1234, אלא 1,234.00. דבר שני, היא מדביקה את סמל המטבע ליד הסכום, וגם זה שירות יפה לאנושות.

בהמשך מודפסות עמודת הכמות ועמודת המחיר. גם את המחיר כדאי להציג בפורמט "מטבעי". העמודה הבאה היא עמודת שם הספר, גם כאן אנו יוצרים קישור. אני מקווה שאתם חושבים עכשיו: היכן ראינו כבר קישור כזה בדיוק? נכון, ברשימת הספרים, תוצאת החיפוש. גם כאן, אנו רוצים להרשות לגולש להגיע לדף המפורט של הספר. לכן אנו יוצרים קישור לדף bookdetail.asp עם פרמטר: book_id.

נסגור את השורה ב- </tr>, נעבור לשורה הבאה וכך הלאה עד לסיום הרשימה.

<u>לסקרנים</u>: מהיכן נלקח סמל המטבע? אולי אני לא רוצה שקלים, אלא דולרים? בלוח הבקרה, הגדרות אזוריות, ניתן לקבוע מה יהיה הסמל! ומי קובע את סמל המטבע, לוח בקרה של השרת או של הלקוח? השרת כמובן. אחרת כל לקוח יראה מטבע אחר, והשמחה תהיה רבה.

סיימנו להציג את כל הספרים בסל הקניות, כעת הגיע הזמן להראות את הסיכום. הקלידו כך, בדף סל הקניות (showbasket.asp), מייד לאחר השורה loop :

```
rs.Close
sql="select sum(total) as sumtotal from basket_list where visitor_id=" &_
session("visitor_id")
rs.Open sql, conn
Response.Write "<tr>"
Response.Write "<td colspan=5 align=center><b>" & formatcurrency(rs("sumtotal"))
Response.Write " : " & "סך הכל" & "</b></td>"
Response.Write "</tr>"
Response.Write "</center></table>"
```

שימו לב, בשורה הראשונה לסגירת הרשימה! פתחנו את הרשימה עם שאילתה מסוימת, כעת אנו רוצים להשתמש בה שוב לשאילתה אחרת, שאילתת הסיכום. **לפני פתיחה – חייבים לסגור!**

בשאילתה אנו משתמשים בפונקציה sum, שמסכמת עבורנו את הסכומים. הבסיס לשאילתה הוא שאילתה אחרת - basket_list שאותה בנינו קודם. כן, מותר לכתוב שאילתה בתוך שאילתה. עשינו זאת, כדי לנצל את חישוב מכפלת המחיר בכמות שמבוצע עבור כל ספר ב- basket_list. כל מה שנותר לנו, הוא לסכם את הסיכומים ולקבל את ה"סך הכל לתשלום".

עוד נקודה מעניינת בשאילתה, היא מתן שם הכינוי (Alias) לתוצאת הסיכום :

select sum(total) **as sumtotal** from basket_list

כך נותנים שם לתוצאת החישוב. השם sumtotal הוא חסר חשיבות, ויכול היה להיות כל מילה אחרת.

פותחים את הרשימה עם השאילתה החדשה, כעת יש בידינו הסכום לתשלום. נותר רק לכתוב אותו, ולידו תווית : "סך הכל". וזהו, סוגרים את הטבלה.

לא נשכח את סגירת הרשימה והקישור. הקלידו :

```
rs.Close
set rs=nothing
conn.Close
set conn=nothing
%>
```

חשוב מאוד לסגור את recordset ואת connection שיצרנו, אחרת לא נצליח לפתוח אותם שוב!

סיימנו להציג את רשימת הספרים וסך הכל לתשלום. כעת אנו רוצים לשלוח את הלקוח לקופה. יש צורך בלחצן, גרפי אם אפשר, שלחיצה עליו תגרום למעבר לדף ההזמנה.

בתיקיה Bookstore\images בתקליטור המצורף תמצאו את התמונה cash.gif :



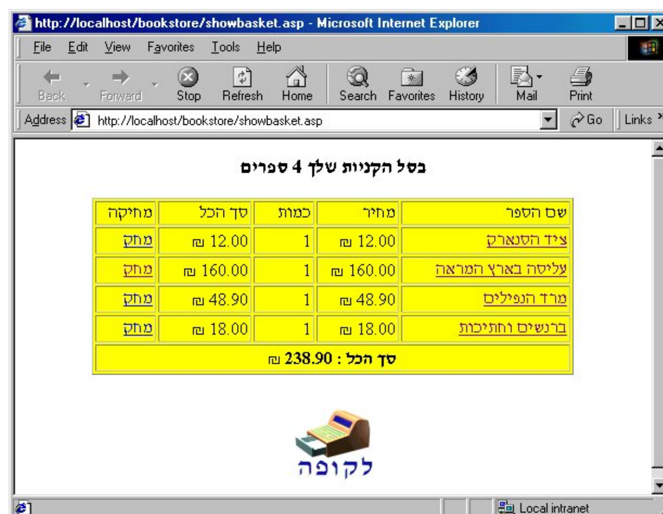
תחילה, נצרף את התמונה לפרויקט שלנו. אחרת היא לא תועתק לשרת!

1. העתיקו את התמונה (למשל, בעזרת הסייר), לתיקיית התמונות שלכם.
2. ב- Project Explorer הציבו את הסמן על התיקיה images, לחצו לחיצה ימנית, ובחרו Add Item. עברו לכרטיסיה Existing והגיעו לתיקיה הנכונה. אם אינכם רואים את הקובץ שלכם, שנו את התיבה Files of Type ל- All Files. כעת אתם רואים את הקובץ. סמנו אותו ואשרו. הקובץ נוסף לתיקיית התמונות שלכם.

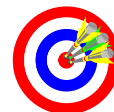
כעת ניצור קישור מהתמונה החדשה לדף ההזמנה order.asp. שימו לב לכך, שסגרנו את קטע ASP, ומעתה אנו ב-HTML.

```
<center>
<A HREF="order.asp"><img src="images/cash.gif" border=0 WIDTH="120"
HEIGHT="80"></A>
</center>
```

בזמן ריצה, הדף צריך להיראות כך :



לחיצה על שם הספר - צריכה להביא אותנו אל דף הספר המלא (bookdetail.asp).
לחיצה על "מחק" - גורמת בשלב זה לשגיאה, מפני שהדף אליו פונים, delfrombasket.asp, עדיין לא כתוב.



תרגול :

מה עם כמויות? חשבו על אפשרויות שונות לתת לגולש אפשרות לשנות את הכמות המוזמנת של כל ספר.

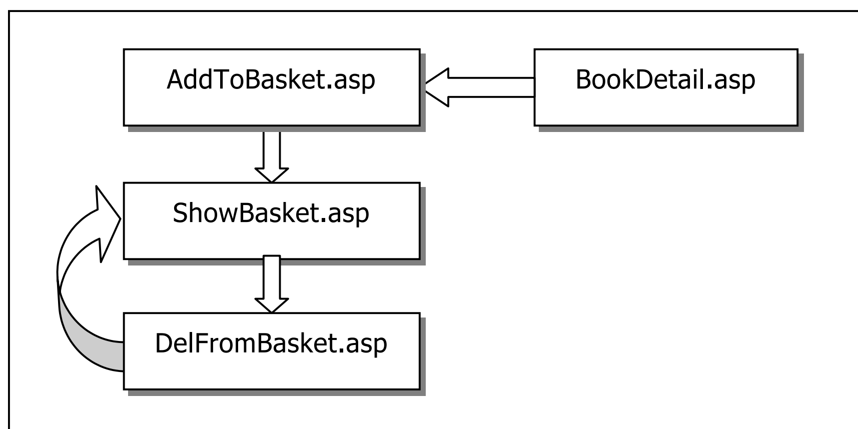
פרק 12

ניהול סל הקניות

מבנה כללי של מנגנון ניהול סל הקניות

לפני ההסבר, הבהרה: סל הקניות המודגם בספר זה אינו תורה מסיני. יש דרכים רבות לממש סל קניות, וחנות מקוונת בכלל. זוהי רק דוגמה לצורך לימוד. העקרונות, אני מאמין, נכונים תמיד, אבל אתם בהחלט מוזמנים לממש אותם מעט אחרת.

התרשים שלפנינו, מיועד לעשות מעט סדר בשלל הדפים, שלוקחים חלק בסל הקניות. הראשון שבנינו, ובוודאי החשוב ביותר, הוא showbasket.asp. תפקידו פשוט: להציג את אותו חלק בטבלת basket, שכידוע משותפת לאלפי גולשים, השייך לגולש היחיד בו אנו עוסקים. ובאותה הזדמנות גם לחשב, בעזרת שאילתה, את סכומי החיוב. התהליך מתחיל בדף פרטי הספר bookdetail.asp. שם החליט הגולש להוסיף ספר מסוים לסל שלו. הוא לוחץ על לחצן, ובעצם, כפי שניתן לראות בתרשים, מפעיל את דף ההוספה - addtobasket.asp. לדף זה אין שום חלק ויזואלי. הגולש לא יראה אותו לעולם! הדף מבצע הוספה, ומייד קורא לדף הצגת הסל. התוצאה: הגולש לחץ על "הוסף לסל הקניות", ולאחר ההוספה, קיבל מייד את הסל שלו. מכאן הוא יכול לחזור לחיפוש, או למחוק ספר. מחיקת ספר מתבצעת רק מתוך סל הקניות. לכל שורה בסל יש לחצן מחיקה. אם נלחץ על לחצן זה, יופעל דף המחיקה delfrombasket.asp (שוב, דף ללא כל חלק ויזואלי, כמו רואה ואינו נראה!), ומייד בסיומו, נשלח שוב showbasket.asp. התוצאה: הגולש לחץ על לחצן המחיקה, ומייד קיבל את הסל המעודכן, ללא הספר שמחק.



בפרק זה, נבנה דפים וקישורים אלה. בסופו יהיה לנו סל קניות עובד, שנוכל להוסיף אליו ולמחוק ממנו ספרים.

הוספת לחצן - הוסף לסל הקניות

הוספת ספר לסל הקניות מתבצעת, באופן טבעי, מתוך מסך הספר הבודד, bookdetail.asp. הגולש ראה את תמונת הספר, את מחירו, והחליט לרכוש אותו. זה המקום לשים את הלחצן הידוע "הוסף לסל הקניות".



עברו לדף bookdetail.asp, והחליטו בעצמכם היכן אתם רוצים את הלחצן. דוגמה ללחצן כזה נמצאת בתקליטור המצורף לספר, בתיקיה bookstore\images (קובץ בשם cart.gif). אתם מוזמנים להשתמש בו, או לעצב משהו יותר מוצלח בעצמכם.

בחרו את הלחצן הרצוי, וצרפו אותו לפרויקט:

1. העבירו אותו פיסית, בעזרת הסייר, לתיקיה images בפרויקט שלכם.
2. ב- Project Explorer סמנו את התיקיה ולחצו לחיצה ימנית, בחרו Add Item, עברו לכרטיסיה Existing, מצאו את התמונה ואשרו. אם אינכם רואים את התמונה שלכם בתיקיה, שנו את סוג התיבה Files of type ל- All Files.

מדוע כל כך חשוב שהפרויקט "יכיר" בתמונה שלכם? כי אחרת הוא לא יעתיק אותה לשרת, התמונה לא תופיע בדף, ואתם תטיחו את ראשיכם במסך בנסיונות להבין מדוע.

היכן ימוקם הלחצן בדף פרטי הספר? על כך תחליטו בעצמכם. אני מזכיר לכם שמכיון שזהו דף מורכב מבחינת המבנה הפיסי, בנינו אותו בשני שלבים: בשלב ראשון, עבדנו

בסביבת ASP, שלפנו את נתוני הספר והכנסנו אותם למשתנים. שלב שני, היה עבודת טבלאות רגילה בה בנינו את הדף תוך שימוש בטבלאות HTML, ושילבנו, במקומות הנכונים, את הנתונים שהכנו מראש. הלחצן שלנו צריך להיות ממוקם בחלק השני, באחד מחלקי הטבלה. לכן, הפקודות שנכתוב כעת הן בעצם HTML רגיל.

הניחו את הסמן במקום בו אתם מעוניינים ליצור את הלחצן, וכעת נבנה קישור פשוט. טוב, לא ממש פשוט, רק עם התחכמות קטנה – שליחת פרמטר. הקלידו :

```
<A HREF=addtobasket.asp?book_id=<%=book_id%>>
<IMG SRC="images/cart.gif" border=0></A>
```

הסבר :

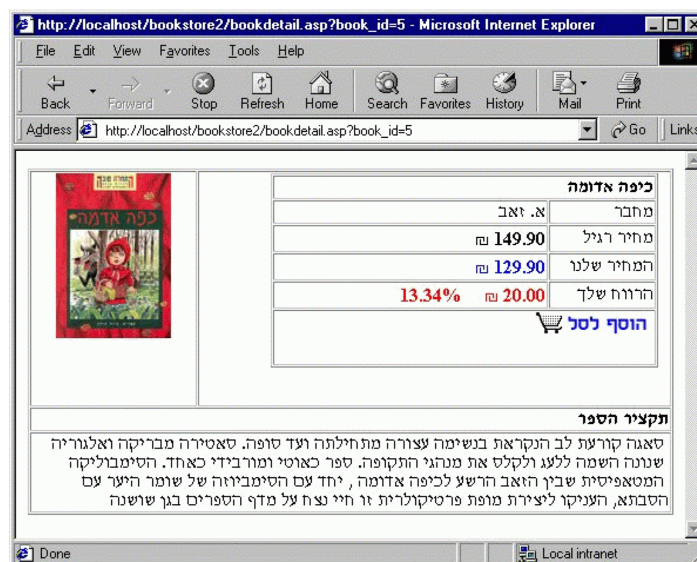
כאשר העברנו נתונים מדף לדף בשיטת GET, נראתה שורת הכתובת כך :

http://localhost/bookstore/get2.asp?fname=Ronen&lname=Almog&submit1=Submit

מייד לאחר שם הקובץ בא סימן שאלה, ואחריו זוגות של פרמטרים. כל שם פרמטר והערך שלו. מותר לנו גם לבנות בעצמנו קריאה כזו. וזה בדיוק מה שאנו עושים. רק שבמקום קוד הספר, אנו כותבים את ערך המשתנה book_id. המשתנה מכיל את קוד הספר הנכון, דאגנו לכך בתחילת הדף.

שימו לב לתוספת : border=0. הנחיה זו תגרום להיעלמות המסגרת הכחולה, המאפיינת קישורים. חשוב לכתוב זאת, מפני שבמקרים רבים המסגרת עלולה לקלקל לכם את העיצוב אם לא התכוונתם שהיא תהיה שם.

דף פרטי הספר - bookdetail.asp, יכול להיראות כך, לאחר הוספת הלחצן :



בכוונה השארתי את מסגרות הטבלאות, כדי שתוכלו לראות את המבנה. לאחר סיום העבודה על הדף, כמובן שבכל הטבלאות נכתוב border=0, כדי להעלים את הקווים.

כשתריצו את הדף (אי אפשר ישירות, צריך לקרוא לדף החיפוש, ולעבור את כל התהליך), הציבו את הסמן על הלחצן. בשורת הסטטוס, למטה, תראו את מסלול הקישור. ודאו שמופיע בו קוד הספר, כך:

http://localhost/bookstore/addtobasket.asp?book_id=1

אם לא מופיע המספר – יש לכם טעות כלשהי. טעות נפוצה היא לשכוח את השווה הקטן לפני המשתנה book_id. בדקו והשוו לספר, עד שיעבוד לכם.

הוספת ספר לסל הקניות

כעת יש לבנות את הדף addtobasket.asp. תפקידו פשוט מאוד: לקבל מספר ספר, להוסיף אותו לסל הקניות של הגולש הנוכחי, ולהציג את סל הקניות.

הוסיפו דף חדש לפרויקט addtobasket.asp. מחקו את כל מה ש- Visual InterDev כתב עבורכם (מדוע? כדי לא להפריע לפקודה Response.Redirect !), והקלידו:

```
<%
dim book_id
book_id=Request.QueryString("book_id")

dim conn
Set conn=Server.CreateObject("ADODB.connection")
conn.Open "dsn=bookstore"

conn.Execute "INSERT INTO basket (book_id, visitor_id) select " & book_id & ", " &_
session("visitor_id")

conn.Close
Set conn=nothing

Response.Redirect "showbasket.asp"
%>
```

הסבר:

שתי השורות הראשונות: קליטת הפרמטר book_id. שימו לב לכך שהקליטה היא בשיטת GET! כיצד יודעים? לפי שכתוב Request.QueryString, ולא Request.Form.

שלוש השורות הבאות הן יצירת Connection סטנדרטי ביותר.

לאחר מכן, מגיעה השורה החשובה ביותר בדף: הוספת שורה לטבלת basket. אנו מפעילים כאן פקודת SQL ישירות על Connection, ללא יצירה מפורשת של Recordset. אנו מוסיפים (insert) - שורה לטבלת basket. שתי העמודות החשובות ביותר בטבלה זו הן מספר הספר, ומספר המבקר. בתור מספר הספר נכניס כמובן את המספר שקיבלנו, book_id. מספר המבקר הוא משתנה ה- Session, שהולך איתנו לאורך כל הדרך, Session("visitor_id"). למי ששכח: זהו "מספר הזהות" של הגולש הנוכחי, והוא האמצעי היחיד להבדיל בינו לבין אלפי גולשים אחרים. מספר זה הוקצה לגולש עם כניסתו לאתר.

הפקודה, שמופיעה בספר בשתי שורות מחוסר מקום, צריכה אצלכם להיות בשורה אחת.

אנו לא שוכחים לסגור את הקישור. לסיום, אנו קוראים לפקודה המוכרת והחביבה Response.Redirect, כדי לגרום לסל הקניות המעודכן לקפוץ אל עיני הגולש. שימו לב לכך, שלדף שכתבנו אין בעצם שום חלק ויזואלי! הוא מבצע את מלאכתו בשקט, וקורא לדף אחר להציג את התוצאה.

שמרו את עבודתכם וסגרו את הדף. כדי לבדוק אם הדף עובד, חפשו ספר, קבלו רשימת ספרים מתאימים לחיפוש, היכנסו לספר מסוים, ולחצו על הלחצן "הוסף לסל קניות". אתם אמורים לראות את סל הקניות עם הספר שסימנתם.

מחיקה מסל הקניות

כעת נכתוב דף ASP, שמוחק ספר מתוך סל הקניות. כזכור לכם, כשבנינו את דף סל הקניות, אחת העמודות בטבלה היתה "מחק". זה בעצם קישור לדף המחיקה, שאותו נבנה עכשיו. הוסיפו לפרויקט דף ASP חדש, וקראו לו delfrombasket.asp. כרגיל בדפי הפניה, מחקו כל זכר ל- HTML, והקלידו:

```
<%
dim book_id
book_id=Request.QueryString("book_id")

dim conn
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"

conn.Execute "DELETE FROM basket WHERE book_id=" & book_id & _
" and visitor_id=" & session("visitor_id")

conn.Close
Set conn=nothing

Response.Redirect "showbasket.asp"

%>
```

כצפוי, דף זה דומה מאוד לדף ההוספה, ההבדלים הם קלים בלבד. גם כאן קליטת הפרמטר היא בשיטת GET (כי הגענו לכאן בעזרת קישור!), לכן הפקודה היא Request.QueryString.

פקודת SQL היא כצפוי פקודת Delete, וגם כאן יש להקפיד למחוק רק את הספרים של המבקר הנוכחי, ולא את כל הספרים במאגר ההזמנות, שהקוד שלהם זהה ל- book_id. שורת המחיקה רשומה בספר בשתי שורות, מחוסר מקום. אצלכם בדף, רשמו אותה בשורה אחת. שמרו את עבודתכם ובדקו. לחיצה על הקישור "מחק" בסל הקניות צריכה לגרום למחיקת השורה, ולהצגה מחדש של סל הקניות עם הספרים שנותרו.



שאלה למחשבה:

נניח שרוצים ליצור לחצן שימחק את כל סל הקניות. כיצד תעשו זאת?

פרק 13

הוספת ניהול כמויות לסל הקניות

סל הקניות שלנו כבר עובד יפה. ניתן להזמין ספרים ולהוסיף אותם לסל, ניתן למחוק ספרים, והסל גם מחשב עבורנו את הסך לתשלום. מה שחסר כעת, הוא מנגנון שבעזרתו יוכל המשתמש לשנות את כמות הספרים המוזמנת. ישנן דרכים רבות לעשות זאת. נראה דרך אחת, פשוטה ויעילה, אך אתם בהחלט מוזמנים לפתח שיטות אחרות.

הרעיון הכללי: את הכמות בדף סל הקניות, נכתוב לא בכתיבה ישירה, כפי שעשינו עד כה, אלא לתוך רכיב TextBox. כך יוכל הגולש "לדרוך" על המספר ולשנותו. יש כאן בעיה קטנה: מה יהיה שמו של הרכיב? הרי יש רבים כאלה (כמספר השורות בסל), ואנחנו לא יודעים כמה יהיו. בעיה נוספת היא, שכתיבה על המסך עדיין לא שינתה את הנתונים בטבלה. גרוע מכך, לא גרמה לחישוב מחדש של מכפלת המחיר בכמות.

לכן, מה שנעשה יהיה Form שקורא לעצמו. כל סל הקניות יהיה בתוך מסגרת של Form, ויהיה לו לחצן "חשב מחדש". לאחר שינוי כמויות, ילחץ המשתמש על לחצן זה, והדף יקרא לעצמו. בקריאה זו לעצמו, ישלח הטופס לעצמו כפרמטרים את כל נתוני הכמויות, בין אם עברו שינוי ובין אם לא. בכניסה לדף יש להחליט: האם זוהי בקשה לחישוב מחדש, או סתם הצגת סל קניות? אם זו בקשה לחישוב, יש לעדכן את הכמויות החדשות. את ההכפלה כמות * מחיר, תעשה השאילתה בעצמה.

כעת לעבודה. לפניכם קטע מתוך דף showbasket.asp עם מספר תוספות. התוספות מודגשות, ויש להוסיף אותן לדף הקיים שלכם (קוד המקור נמצא ב-showbasket3.asp).

ראשית, המצאנו משתנה בשם counter.

שנית, בתוך לולאת הכתיבה של נתוני סל הקניות, החלפנו את השורה שהציגה את השדה quantity בשורות חדשות, המציגות את אותו שדה בתוך תיבת טקסט.

כעת שימו לב לטריק: מכיון שאין לנו מושג כמה שורות עומדות להיות בסל הקניות, קשה לנו לתת שמות לתיבות הטקסט. לכן ניצור להן שמות באופן דינמי. שמה של כל תיבה יהיה Q (לזכר Quantity), ועוד מספר המונה counter, שמתקדם כל שורה באחד. כך התיבה בשורה הראשונה תהיה Q1, בשנייה Q2 וכן הלאה.

```
counter=1
do while not rs.EOF
 Response.Write "<tr>"

 Response.Write "<td  align=right>"
 Response.Write "<A HREF=" & "delfrombasket.asp?book_id=" & rs("book_id") & ">"
 Response.Write "מחק" & "</A></td>"& vbcrlf
 Response.Write "<td align=right>" & formatcurrency(rs("total")) & "</td>"& vbcrlf
 Response.Write "<td align=right><input type=text name=Q" & counter & " value=" &_
   rs("quantity")
 Response.Write " style=""HEIGHT: 22px; WIDTH: 30px"""& "></td>"& vbcrlf
 Response.Write "<td align=right>" & formatcurrency(rs("price")) & "</td>"& vbcrlf
 Response.Write "<td align=right><A HREF=" & "bookdetail.asp?book_id="
 Response.Write rs("book_id") & ">"
 Response.Write rs("title") & "</A></td>"& vbcrlf

 Response.Write "</tr>"
 counter=counter+1
 rs.MoveNext
loop
```

הצגנו את ערכי הכמות בתוך תיבות טקסט, וזהו צעד יפה לכיוון המטרה. כעת נבנה Form סביב תיבות טקסט אלו. ממש מעל הגדרת המונה Counter, ניצור Form :

```
Response.Write "<FORM action=""showbasket.asp"" method=POST id=form2
name=form2>"
```

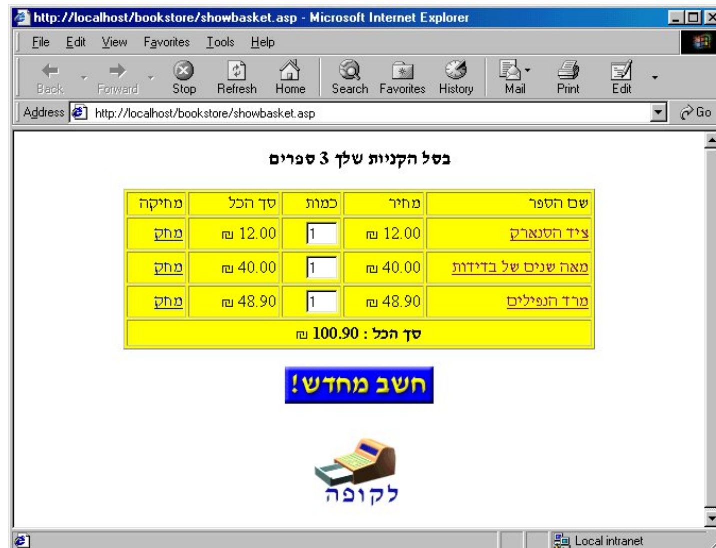
נכון, זה טופס משונה, שקורא לעצמו.

לא נשכח לסגור את הטופס, ובחלק התחתון, לאחר סגירת הלולאה, נכתוב </Form>.

כעת כל מה שנחוץ, הוא לחצן Submit. ניתן לגרור לחצן כזה מארגז הכלים, אבל אנחנו כבר יותר מתוחכמים, ורוצים לחצן משלנו. לכן נצרף לפרויקט את הלחצן recalc.gif, למדור התמונות images, ונוסיף לפני סגירת הטופס את השורה :

```
<center><input type="image" SRC="images/recalc.gif" value="Submit" id="submit1"
name="submit1" WIDTH="140" HEIGHT="35"></center>
```

כעת הדף שלנו ייראה בערך כך :



כעת עלינו לדאוג לכך, שהלחצן אכן יגרום לחישוב מחדש של המחירים ושל הסך הכל.

תחילה, יש להודיע לדף המקבל (הדף שלנו...) כמה שורות יש לנו. כלומר: כמה ערכי כמות עליו לקלוט. לכן נוסיף לטופס, ממש לפני סגירתו, את השורה הבאה:

```
<input type="hidden" name="counter" value="<%=counter%>">
```

זהו משתנה חבוי (hidden), המכיל את ערך המונה (מספר השורות שכתבנו).

כעת נעבור לתחילת הדף. כאשר דף זה עולה עליו להחליט: האם אני מוצג באופן רגיל, כסל קניות, או שזהו חישוב מחדש של כמויות? את זאת הוא יידע לפי הפרמטר counter. אם הוא אפס או ריק – סימן שזו קריאה רגילה. אם יש בו ערך מספרי – זהו חישוב מחדש.

נוסיף את השורות הבאות בתחילת הדף:

```
counter=Request.Form("counter")
if counter<>"" then     ' recalculation is needed
counter2=1

do while not rs.EOF
  if not isnumeric(Request.Form(counter2)) or Request.Form(counter2)="0" then
    conn.Execute "delete from basket where basket_id=" & rs("basket_id")
  elseif  rs("quantity")<>Request.Form(counter2) then
    conn.Execute "update basket set quantity=" & Request.Form(counter2) &_
         " Where basket_id=" & rs("basket_id")
  end if
```

```
 counter2=counter2+1
 rs.MoveNext
loop
 rs.close
 rs.Open sql, conn ' Open the rs again, data has changed!
end if
```

הסבר:

1. קלוט את הפרמטר Counter מהדף שקרא לך.
2. רק אם הוא שונה ממחרוזת ריקה... (כלומר, זהו חישוב כמויות מחדש).
3. אם הכמות החדשה היא אפס או שאינה מספרית כלל (מחרוזת ריקה או כל "זבל" אחר), מחק את השורה. הגולש כנראה רוצה לבטל שורה זו.
4. אם הכמות החדשה שונה מהכמות הישנה, עדכן את הכמות לכמות החדשה.
5. סגור את Recordest ופתח אותו מחדש. זה יגרום ל"ריענון" של הרשימה.

יש צורך בכך, מפני שהנתונים השתנו. כך אנו קוראים מהשאילתה את הנתונים המעודכנים, גם לגבי הכמויות ששתנו, וגם לגבי עמודת "סך הכל". שימו לב, בשום מקום לא חישבנו את הסך הכל החדש! השאילתה תעשה זאת עבורנו.

כך ייראה סל הקניות שלכם, לאחר לחיצה על הלחצן "חשב מחדש!"

http://localhost/bookstore/showbasket.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit

Address http://localhost/bookstore/showbasket.asp Go

בסל הקניות שלך 3 ספרים

שם הספר	מחיר	כמות	סך הכל	מחיקה
ציד הסנארק	12.00₪	3	36.00₪	מחק
מאה שנים של בדידות	40.00₪	2	80.00₪	מחק
מרד הנפילים	48.90₪	1	48.90₪	מחק
סך הכל : 164.90₪				

חשב מחדש!

לקופה

http://localhost/bookstore/bookdetail.asp?book_id=10 Local intranet

מה פירוש Request.Form(counter2)? האם זה לא נראה לכם קצת מוזר? האם לא חסרות כאן מרכאות?

התשובה היא לא. Request.Form הוא אוסף (Collection), וככזה, מותר לפנות אל איבריו בשמם הפרטי, לדוגמה ("Request.Form("fname, אך מותר לפנות אליהם גם לפי **מספרים**. לדוגמה: (Request.Form(2 נותן את הפרמטר השני. counter2 רץ מ- 1 עד מספר השורות בסל, ובעזרתו אנו שולפים את הכמויות שהוזנו. כזכור לכם, תיבות הטקסט המכילות את הכמויות נשלחות אלינו כפרמטרים. שמותיהן ידועים לנו: Q1, Q2 וכן הלאה. אבל למען הנוחות בחרתי הפעם להשתמש בשיטה זו.

לשיקול דעתכם:

כעת, כשיש לנו ניהול כמויות משוכלל, ניתן לוותר על העמודה "מחק". הגולש יכול לכתוב כמות אפס, לחשב מחדש, והשורה תיעלם.

לסיכום, קוד הדף כולו (showbasket3.asp):

```
<%
dim conn
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"

dim rs, sql
set rs=Server.CreateObject("ADODB.recordset")
sql="select * from basket_list where visitor_id=" & session("visitor_id")

rs.Open sql, conn, 3  ' Static, so we can count the results.

counter=Request.Form("counter")
if counter<>"" then  ' recalculation is needed
counter2=1

do while not rs.EOF
  if not isnumeric(Request.Form(counter2)) or Request.Form(counter2)="0" then
    conn.Execute "delete from basket where basket_id=" & rs("basket_id")
  elseif  rs("quantity")<>Request.Form(counter2) then
    conn.Execute "update basket set quantity=" & Request.Form(counter2) &_
    " where basket_id=" & rs("basket_id")
  end if

 counter2=counter2+1
 rs.MoveNext
loop
 rs.close
 rs.Open sql, conn ' Open the rs again, data has changed!
end if
```

```
bookcount=rs.RecordCount

if rs.EOF =true and rs.BOF=true then  ' no book found
  Response.Redirect "basketisempty.asp"
end if

' Show menu only after (possible) redirection.
' We'll reach menus only in chapter 18.
' You can comment out these 2 lines, return them later.
session("menu_id")=2%>
<!-- #include file = "menu.asp" -->

<%
Response.Write "<center><h3>" & "בסל הקניות שלך" & "  "
Response.Write bookcount & "  " & "ספרים" & "</h3>"
Response.Write "</center>"


Response.Write "<FORM action=""showbasket.asp"" method=POST id=form2 "
Response.Write " name=form2>"& vbcrlf
Response.Write "<center><table border=1 cellpadding=2 BGCOLOR=Yellow>"
Response.Write "<tr>"
Response.Write "<td align=right width=50>מחיקה</td>"& vbcrlf
Response.Write "<td align=right width=80>סך הכל</td>"& vbcrlf
Response.Write "<td align=right width=50>כמות</td>"& vbcrlf
Response.Write "<td align=right width=70>מחיר</td>"& vbcrlf
Response.Write "<td align=right width=150>שם הספר</td>"& vbcrlf
Response.Write "</tr>"
counter=1
do while not rs.EOF
 Response.Write "<tr>"

 Response.Write "<td  align=right>"
 Response.Write "<A HREF=" & "delfrombasket.asp?book_id=" & rs("book_id") & ">"
 Response.Write "מחק" & "</A></td>"& vbcrlf
 Response.Write "<td align=right>" & formatcurrency(rs("total")) & "</td>"& vbcrlf
 Response.Write "<td align=right><input type=text name=Q"
 Response.Write counter & " value=" & rs("quantity")
 Response.Write " style=""HEIGHT: 22px; WIDTH: 30px"""& "></td>"& vbcrlf
 Response.Write "<td align=right>" & formatcurrency(rs("price")) & "</td>"& vbcrlf
 Response.Write "<td align=right><A HREF=" & "bookdetail.asp?book_id="
 Response.Write rs("book_id") & ">"
 Response.Write rs("title") & "</A></td>"& vbcrlf
```

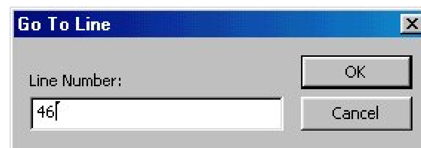
```
 Response.Write "</tr>"
 counter=counter+1
 rs.MoveNext
loop
rs.Close
sql="select sum(total) as sumtotal from basket_list where visitor_id=" &_
    session("visitor_id")
rs.Open sql, conn
Response.Write "<tr>"
Response.Write "<td colspan=5 align=center><b>" &_
  formatcurrency(rs("sumtotal")) & "  :  " & "סך הכל " & "</b></td>"
Response.Write "</tr>"
Response.Write "</center></table>"
rs.Close
set rs=nothing
conn.Close
set conn=nothing
%>
<br>
<input type="hidden" name="counter" value="<%=counter%>">
<center><input type="image" SRC="images/recalc.gif" value="Submit"
  id="submit1" name="submit1" WIDTH="140" HEIGHT="35"></center>
</form>

<center>
<a HREF="order.asp"><img src="images/cash.gif" border="0" WIDTH="120"
  HEIGHT="80"></a>
</center>
```



טיפ: קפיצה לשורה

לעיתים אתם רוצים לקפוץ לשורה מסוימת בדף. למשל, כאשר יש שגיאה בשורה מסוימת. בשורת הסטטוס, בתחתית המסך, כתוב מיקום הסמן כרגע. לדוגמה: Col 14 Ln 23. לחיצה כפולה על מספר השורה תפתח את החלון הבא:



הקישו את מספר השורה אליה רוצים להגיע, ואשרו. הסמן יקפוץ אל השורה המבוקשת. לאותו חלון ניתן להגיע גם בלחיצה על Ctrl+G.

פרק 14

ביצוע הרכישה בפועל

דף הזמנה

הגענו לרגע האמת. הגולש חיפש ספרים, ראה תמונות, השווה מחירים, הוסיף לסל הקניות, הוריד מסל הקניות, והנה הוא כאן, נחוש בדעתו לבצע רכישה. לדף שניצור בפרק זה מגיע הגולש כתוצאה מלחיצה על הלחצן "לקופה" בסל הקניות. בעצם זהו דף קלט פשוט, שיש לו כמה מטרות:

1. לקלוט פרטים מלאים על הלקוח, כולל כתובת למשלוח.
2. לקלוט פרטי כרטיס אשראי לחיוב, כולל תוקף.
3. להציג בפניו שוב את הסכום לתשלום, לבדיקה סופית.

כבר כתבנו דפי קלט, ואף יותר משיטה אחת. בדף זה נלמד טכניקה נוספת: דף קלט המוצמד לנתונים. כלומר: אם נתון כלשהו, מהנתונים אותם נשאל את הגולש שלנו, כבר ידועים לנו, נמלא אותם בעצמנו. לדוגמה, אם הוא הצטרף למועדון, ומילא את פרטיו, אז שמו וכתובתו כבר ידועים לנו. נכתוב אותם בשדות המתאימים ונקל עליו את מלאכת הקנייה. רק לגבי כרטיס האשראי ננהג זהירות: אולי לא הוא יושב מול המחשב, אלא בן משפחה או עובד אחר באותו משרד? לכן גם אם אנחנו יודעים את מספר הכרטיס, ניתן לו להקליד אותו שוב, כאמצעי אבטחה.

החלון צריך להיראות כך :

אין ספק שזהו אחד הדפים חסרי החן שנראו כאן לאחרונה. הטבלה קיבלה עובי גבול 1 רק כדי שתראו את המבנה, רצוי להעלים אותה. אתם מוזמנים לשפר את העיצוב ולצבוע בצבעים עליזים.

ולעניין. בטבלה אנו רואים את כל פרטי הלקוח, בתוספת פרטים הנדרשים על ידי חברות האשראי : סוג כרטיס, מסי כרטיס תוקף (חודש ושנה) ומספר זהות.

בשורה התחתונה ניתן לראות גם את הסכום לחיוב, לאישור סופי. לחיצה על "המשך" תגרום לביצוע הפעולה.

הוסיפו לפרויקט שלנו דף חדש : order.asp, ונתחיל לעבוד.

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>
<center><STRONG>פרטי התשלום</STRONG></center>
<%
dim conn
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"

dim rs, sql
set rs=Server.CreateObject("ADODB.recordset")
sql="select * from visitors where visitor_id="& session("visitor_id")
rs.Open sql, conn
```

```
dim visitor_name, email, address, phone, credit_card_id
dim credit_card_num, credit_card_exp_date
visitor_name=rs("visitor_name")
email=rs("email")
address=rs("address")
id=rs("id")
phone=rs("phone")
credit_card_id=rs("credit_card_id")
credit_card_num=rs("credit_card_num")
credit_card_exp_month=month(rs("credit_card_exp_date"))
credit_card_exp_year=year(rs("credit_card_exp_date"))

rs.close

%>
```

חלק ראשון - קריאת נתוני הלקוח מתוך הטבלה

ניתן לראות בקוד יצירת קישור רגיל ופתיחת רשימה. השאילתה פשוטה מאוד: תן לי את כל פרטי הלקוח, שהקוד שלו שווה ל- Session("visitor_id"). כלומר, הלקוח הנוכחי.

לאחר מכן, מתחילה קריאה של כל המידע מהטבלה לתוך משתנים. אלה ישמשו אותנו בחלק הבא.

שימו לב להתחכמות קלה: תאריך התוקף נשמר כשדה תאריך. לכן, אם רוצים לחלץ ממנו את החודש או השנה, ניתן להשתמש בפונקציות year, month, שתוצאתן מספר פשוט.

חלק שני - יצירת Form

חלק זה תעשו בעצמכם: יצירת Form, עם כל שדות הקלט בתוכו. תחילה, הכריזו כרגיל על טופס:

```
<FORM action="orderproc.asp" method=POST id=form1 name=form1>
```

אחר כך צרו טבלה (עובי 1, כדי שתראו מה אתם עושים):

```
<center> <TABLE WIDTH=350 BORDER=1>
```

כל שורה בטבלה תכיל שתי עמודות: הימנית היא כותרת, והשמאלית שדה קלט.

```
<TR><TD align=right><INPUT id=text1 name=visitor_name
value="<%=visitor_name%>" dir="rtl"></TD>
<TD>שם מלא</TD></TR>
```

שימו לב להתחכמות: "value="<%=visitor_name%>. הערך שאנו מכניסים לשדה הוא אותו ערך, שקראנו בחלק הראשון מתוך הטבלה. זה בדיוק מה שיגרום לאפקט הרצוי – שדה קלט ממולא אוטומטית. חזרו על טכניקה זו עבור כל השדות.

הביטוי "dir="rtl יגרום לסמן להתייצב בצד ימין של תיבת הטקסט, כמקובל בעברית.

כדאי להתעכב על השדה בו הלקוח בוחר כרטיס אשראי. זו הזדמנות להתאמן בטריק שלמדנו בפרק הקודם: כיצד למלא תיבת קומבו (Combo Box) מתוך טבלה.

cards : Table

card_name	card_id
Visa	1
IsraCard	2
American Express	3
Diners	4
	0

1 of 4

תחילה, כדאי שתהיה לנו טבלה. צרו במסד הנתונים שלכם טבלת כרטיסי אשראי. יהיו בה קוד כרטיס ושם כרטיס. הכניסו מספר רשומות. לדוגמה: 1 – ויזה, 2 – ישראכרט, 3 – אמריקן אקספרס וכדומה.

כעת נתחיל לכתוב. בשורה הראשונה אנו יוצרים את רכיב הקלט (זה קל!).

בשורה השנייה אנו עוברים ל-ASP, ופותחים רשימה (קישור יצרנו בתחילת הדף). השאילתה פשוטה מאוד: קרא את כל טבלת כרטיסי האשראי. מתחילים בסריקה של כל שורות הרשימה, וכל עוד לא הגענו לסוף... עבור כל שורה ברשימה (בעצם, סוג כרטיס אשראי) – כתוב לדף, בעזרת Response.Write, שורת <OPTION>.

כעת, שימו לב להתחכמות: אם הערך הנקרא זהה למשתנה, שקראנו בחלק א (במילים אחרות: אם זה כרטיס האשראי, שידוע לנו ששייך לגולש!), אז הוסף לקוד את המילה selected. מילה קטנה זו תגרום לערך זה להיות "נבחר" בתיבת הקומבו.

בין אם נבחר ובין אם לא, הוסף את פרטי הכרטיס, כדי לסיים את השורה <OPTION>.

הקוד הוא:

```
<TR><TD align=right><SELECT id=select1 name=credit_card_id dir="rtl">
<%
sql="select * from cards order by card_id"
rs.Open sql, conn
while not rs.EOF
 Response.Write "<OPTION"
  if rs("card_id")=credit_card_id then
    Response.Write " selected "
  end if
 Response.Write "value=" & rs("card_id") & ">" & rs("card_name") &_
"</OPTION>" & vbcrlf
 rs.MoveNext
wend

rs.Close
%>
 </SELECT></TD>
<TD>כרטיס אשראי</TD></TR>
```



הערה:

הקפידו על רווח לפני ואחרי המילה selected, אחרת היא תיצמד למילים הסמוכות לה.

הביטוי "dir="rtl גורם לקומבו להיות מיושר לימין, כאשר החץ שלו בצד שמאל, כמקובל בעברית.

עכשיו, כשכל הסודות בידיכם, צרו גם טבלת שנים וטבלת חודשים, וקלטו מהלקוח את תאריך תוקף כרטיס האשראי שלו. צריכות להיות שתי תיבות קומבו (Combo Boxs): אחת לשנים והשנייה עבור החודשים.

נעבור לחלק הבא: הצגת הסכום לתשלום. גם זה די פשוט: כל מה שנדרש כאן, זו שאילתת סיכום על סל הקניות. ההגבלה היא כמובן לפי מספר הלקוח - visitor_id.

```
<%
dim grandtotal
sql="select * from basket_list where visitor_id=" & session("visitor_id")
rs.Open sql, conn
if rs.EOF =true and rs.BOF=true then
   grandtotal=0
else
rs.Close
sql="select sum(total) as grandtotal from basket_list where visitor_id=" & _
session("visitor_id")
rs.Open sql, conn
grandtotal=rs("grandtotal")
end if
%>
```

גם בהצגת הסכום, ישנה התחכמות קטנה שראויה להסבר.

את הסכום לתשלום לא נציג בתיבת קלט, מפני שלא נרצה שהגולש ישנה את הסכום, אלא סתם נדפיס אותו לדף. אבל, אנחנו צריכים רכיב קלט, כדי להעביר את הנתון לדף הבא! לכן, נשתמש ברכיב נסתר - hidden, ולתוכו נכניס את אותו סכום. הנה כך:

```
<TD align=right><input type=hidden value=<%=grandtotal%> name="grandtotal">
<%=formatcurrency(grandtotal,2,true)%></TD>
<TD>לתשלום</TD>
```

לסיום, לאחר סגירת הטבלה, נציג לחצן גרפי שישלח אותנו לדף הבא:

```
<input type="image" SRC="images/goon.gif" value="Submit" id="submit1"
name="submit1" >
```

כתבו זאת, ונעבור לדף הבא.

דף קבלת ההזמנה

קחו הרבה אוויר, זה הולך להיות אחד הדפים המורכבים בפרויקט כולו.

צרו דף חדש, בשם orderproc.asp. זהו השם שכתוב ב- action של דף ההזמנה, לכן נגיע לכאן, כאשר הגולש יאשר את ההזמנה.

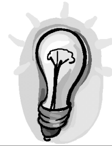
מחקו את שלד ה-HTML שיצר עבורכם Visual InterDev. אין בו צורך.

משימה ראשונה - קליטת פרמטרים

לקלוט את כל הפרמטרים, שנשלחו לנו על ידי הדף הקודם. הקלידו:

```
dim visitor_name, email, address, phone, credit_card_id
dim credit_card_num, credit_card_exp_date, grandtotal
dim sql
visitor_name=Request.Form("visitor_name")
email=Request.Form("email")
address=Request.Form("address")
id=Request.Form("id")
phone=Request.Form("phone")
credit_card_id=Request.Form("credit_card_id")
credit_card_num=Request.Form("credit_card_num")
credit_card_exp_month=Request.Form("credit_card_exp_month")
credit_card_exp_year=Request.Form("credit_card_exp_year")
grandtotal=Request.Form("grandtotal")
```

הסבר: המצאנו מספר משתנים, וקלטנו לתוכם ערכים. הערכים נשלחו בשיטת POST, ולכן כתבנו Request.Form.



טיפ:
אתם יכולים להמציא שמות משתנים לפי רצונכם, אבל כדאי מאוד לתת להם בדיוק את אותם שמות כמו לשדות הטבלה. כך יהיה לכם פחות לזכור, ותטעו פחות!!!

משימה שנייה - בדיקת "כשרות"

קיבלנו שדות, ואנו עומדים להכניס אותם למסד הנתונים. כדאי מאוד לערוך לפני כן בדיקה קטנה.

האם הנתונים הוקלדו במלואם? האם הם חוקיים? כל שדה יכול להיבדק בנפרד, והבדיקות יכולות להיות שונות לגמרי זו מזו. בדוגמה שלנו, נבדוק רק שלא נשכחו פרטים. בחנות אמיתית אולי נבדוק את מספר כרטיס האשראי כנגד "קובץ חסומים" (יש חיה כזו, וניתן לקבל אותה מחברות האשראי), נבדוק ספרת ביקורת וכך הלאה.

הקלידו:

```
if visitor_name = nil or email=nil or id=nil or phone=nil or credit_card_id=nil or
credit_card_num=nil then
        Response.Write "<center><h2>"
        Response.Write "חלק מהפרטים שהקלדת חסרים או שגויים"
        Response.Write "</h2></center>"
        Response.Write "<center><h2>"
        Response.Write "חזור לדף הקודם, תקן, ובצע שנית" & ""
        Response.Write "</h2></center>"
else
```

בשורה הראשונה, כל הבדיקה צריכה להיות בשורה אחת.

הסבר: בדקנו ב-if אחד ארוך, אם אחד השדות ריק (nil). אם כן, כתוב הודעה מתאימה לגולש. אם לא... המשך הלאה. זכרו כי חסר end if לסגירת המעגל.

זו דוגמה לבדיקת תקינות פשוטה. מומלץ לבדוק כל שדה בנפרד ולהציג הודעה מתאימה למשתמש אם יש בעיה בקלט של אותו שדה. באופן שכזה, ידע המשתמש מה לא בסדר וכיצד עליו לנהוג.

משימה שלישית - עדכון פרטי הגולש

אנו רוצים לשמור את פרטי הגולש אצלנו, לצורך שימושים עתידיים. לכן, נבנה שאילתת עדכון, המעדכנת את רשומת הגולש בטבלה visitors. זו אינה שאילתת הוספה, מפני שלגולש שלנו כבר נוצרה שורה בטבלה. כל מה שצריך הוא, רק עדכון פרטים. הקלידו:

```
dim conn
set conn=Server.CreateObject("ADODB.connection")
conn.open "dsn=bookstore"

sql="UPDATE visitors SET visitor_name='" & visitor_name
sql=sql & "', email='" & email
sql=sql & "', address='" & address
sql=sql & "', id='" & id
sql=sql & "', phone='" & phone
sql=sql & "', credit_card_id='" & credit_card_id
sql=sql & "', credit_card_num='" & credit_card_num
sql=sql & "', credit_card_exp_date=#" & credit_card_exp_month
sql=sql & "/01/" & credit_card_exp_year & "#"
sql=sql & " where visitor_id=" & session("visitor_id")

conn.Execute sql
```

הסבר: בשורות הראשונות אנו יוצרים קישור רגיל.

משפט Update בשפת SQL בנוי כך:

```
UPDATE <table> SET field1=value1, field2=value2, …
```

כלומר, רשימה ארוכה של זוגות: שם שדה – והערך המתאים לו, שם שדה – ערך וכך הלאה.

כדי לא לכתוב שורה אחת באורך קילומטר, נהוג לבנות את משתנה המחרוזת בצורה הדרגתית.

שם המשתנה, אגב, לא חייב להיות sql. זה סתם הרגל ישן. השורה:

```
sql=sql & "', id='" & id
```

פירושה: לתוך המשתנה sql, הכנס את ערכו הנוכחי של sql ועוד משהו.

כלומר, הוסף את המשהו למשתנה sql.

כך הולך המשתנה sql, ונבנה משורה לשורה.



טיפ חשוב:

עבור שדה המוגדר String, יש להקיף את הערך בגרשים. עבור ערך מספרי – לא. לדוגמה:

email='bill@microsoft.com', credit_card_id=2

בנושא זה לא קובעת מהות השדה, אלא כיצד השדה מוגדר בטבלה! לדוגמה, שדה id מוגדר אצלנו כטקסט (אתה יכול לחשוב, בטעות, ששדה id הוא מספר, אבל מה שקובעת זו הגדרת השדה בטבלה), לכן הוא מקבל גרש לפניו וגרש אחריו. אחרת – לא עובד!

לאחר בניית השאילתה, היא מופעלת כנגד הקישור שבנינו. אם אין שגיאה, היא עובדת.

איך בודקים מספר כרטיס אשראי?

האלגוריתם לבדיקת מספרי כרטיסי אשראי נקרא mod 10 או LUHN Formula, והוא פתוח לכל. כעת תדעו אותו גם אתם ותוכלו לבדוק בעזרתו את נכונות הכרטיסים באתר שלכם.

מספרי הכרטיסים הם בדרך כלל בעלי 13 עד 16 ספרות. במהלך הבדיקה אנו מתעלמים מרווחים, מקפים ושאר תוספות. נתחיל בתיאור שלבי הבדיקה, ונסיים בדוגמה.

1. ראשית, כתוב את כל הספרות על דף. לצורך החישוב הידני ניעזר בטבלה.
2. מתחת לכל ספרה רשום תוצאת מכפלה: החל מהספרה **הימנית** ביותר, ספרה שמקומה אי זוגי – הכפל ב- 1, כלומר רשום את הספרה עצמה. אם המקום זוגי – הכפל ב- 2.
3. אם תוצאת המכפלה גדולה מ- 9 – סכם את סכום הספרות. לדוגמה, אם התוצאה 14, רשום 5, מפני ש- 1+4=5.
4. סכם את המספרים שהתקבלו. אם התוצאה מתחלקת ב- 10 (ספרה אחרונה 0) – המספר כשר. אם לא – המספר שגוי.

דוגמה לבדיקת כרטיס ויזה:

2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	
4	5	8	0	1	2	3	4	5	6	7	8	9	0	1	1	מספר מקורי
8	5	16	0	2	2	6	4	10	6	14	8	18	0	2	1	מכפלה ב- 1 או 2
8	5	7	0	2	2	6	4	5	6	5	8	9	0	2	1	סכומי ספרות
8	13	20	20	22	24	30	34	39	45	50	58	67	67	69	**70**	סכום מצטבר

הסכום שהתקבל הוא 70, הוא מתחלק ב- 10, ולכן המסקנה שהמספר תקין.

מובן שהעובדה שמספר הכרטיס עובר את הבדיקה, אינה אומרת שהכרטיס בתוקף, שאינו גנוב, וכן הלאה. זו בדיקה פשוטה שחוסמת מתחזים ברמה נמוכה ומונעת טעויות הקלדה.

ידועים עוד מספר מאפיינים על סוגי הכרטיסים השונים:

ויזה: אורך המספר הוא 13 או 16 ספרות (בארץ רק 16). המספר תמיד יתחיל ב- 4.

דיינרס: אורך 14 ספרות, מתחיל ב- 30, 36, או 38.

אמריקן אקספרס: אורך 15 ספרות, מתחיל ב- 34 או 37.

מאסטר-כרד: אורך 16 ספרות. מתחיל ב- 51 עד 55.

ישראכרט: אורך 8 ספרות, ובכמה מקרים שבדקתי – לא עובר את הבדיקה הנ"ל.

בעזרת מידע זה, אתם יכולים לא רק לבדוק את תקינות המספר, אלא גם "לנחש" לפי המספר את סוג הכרטיס.

משימה רביעית - יצירת הזמנה

הגיע הזמן לדון במבנה ההזמנה. כאשר רצינו לשמור במסד הנתונים אובייקטים שונים, יצרנו עבורם טבלה: טבלת מבקרים, טבלת ספרים וכך הלאה. אך הזמנה (גם חשבונית מס, תעודת משלוח, קבלה ועוד) אינה אובייקט פשוט, שניתן לייצגו בטבלה אחת. דרושות כאן שתי טבלאות: טבלה ראשית ומשנית. בשפת מנהלי מסד נתונים: טבלת אב וטבלת בנים.

נתבונן בהזמנה טיפוסית:

ספרוני – מסחר אלקטרוני בע"מ
רח' מנדלי מוכר ספרים 77 תל אביב

<u>הזמנה מס' 12345</u>

לכבוד: תאריך: 2.7.2000
מוישה זוכמיר
לבונטין 44
תל אביב

שם הספר	**מחיר**	**כמות**	**סך הכל**
עליסה בארץ המראה	50	2	100
ציד הסנארק	60	1	60
סך הכל:			160.00
מע"מ 17%:			27.20
לתשלום:			**187.20**

כיצד נשמור ״חיה״ כזאת בטבלה?

הבעיה העיקרית היא השורות. אין לנו שום מושג כמה שורות עשויות להיות בהזמנה. לכן, נפריד בין שדות קבועים שיהיו ללא ספק בהזמנה, לבין שדות השייכים לשורות. שדות ההזמנה יהיו למשל: מספר הזמנה, תאריך, שם (או קוד) הלקוח, סך הכל, סכום המע״מ וסך הכל לתשלום. אם אתם גובים תשלום עבור משלוח, זה המקום לציין אותו, ומובן שה״סך הכל לתשלום״ צריך לכלול אותו.

עד כאן שדות ההזמנה. בטבלה נפרדת, ננהל את שורות ההזמנה. אילו שדות יהיו בטבלת שורות ההזמנה? תחילה, מספר ההזמנה. זו השורה החשובה ביותר, בלעדיה לא נדע לאיזו הזמנה שייכת כל שורה! לאחר מכן, נכתוב שדות המתארים את השורה: קוד הספר, מחירו, כמות וסך הכל.



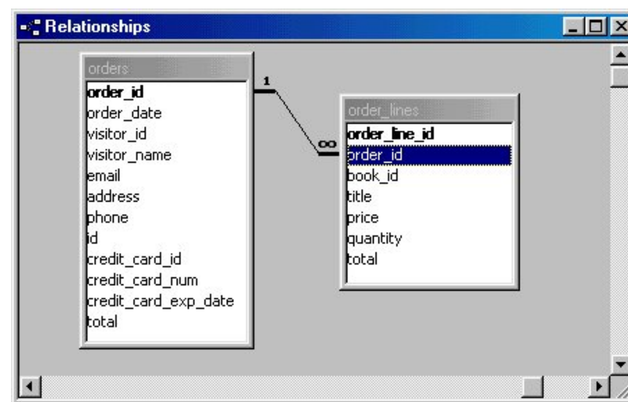
שאלה: למה לכתוב מחיר? הרי תמיד אני יכול להביא אותו בשאילתה מטבלת הספרים!

תשובה: המחיר בטבלה, כמו בכל חנות, עשוי להשתנות. מרגע שלקוח ביצע הזמנה – אין לגעת בסכומים! בין אם עלה המחיר ובין אם ירד, הלקוח כבר קנה, והעניין סגור.

מסיבה זו, נעתיק לפרטי ההזמנה גם את שמו, כתובתו ופרטי כרטיס האשראי של המזמין.

איננו סומכים על טבלת Visitors, מדוע? מפני שהגולש יכול לשנות בה את הנתונים! אל ההזמנה לא תהיה לו גישה לעולם.

היחסים בין שתי הטבלאות החדשות, יהיו כך: עבור כל שורה בטבלת הזמנות, יכול להיות מספר בלתי מוגבל של שורות בטבלת שורות הזמנה.



בכל שורה כזאת יש מספר הזמנה. בעזרתו ניתן למצוא בקלות את כל השורות השייכות להזמנה מסוימת.

בנו את טבלת הזמנות (orders), ואת טבלת שורות הזמנה (order_lines), לפי הטבלאות הבאות:

טבלת הזמנות - Orders

orders : Table

Field Name	Data Type	Description
order_id	AutoNumber	
order_date	Date/Time	
visitor_id	Number	
visitor_name	Text	
email	Text	
address	Text	
phone	Text	
id	Text	
credit_card_id	Number	
credit_card_num	Text	
credit_card_exp_date	Date/Time	
total	Number	

Field Properties

General | Lookup

Format	General Date
Input Mask	
Caption	
Default Value	Now()
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (Duplicates OK)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

טבלת שורות הזמנה - Order_lines

order_lines : Table

Field Name	Data Type	Description
order_line_id	AutoNumber	
order_id	Number	
book_id	Number	
title	Text	
price	Number	
quantity	Number	
total	Number	

Field Properties

General | Lookup

Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

כעת, כשהטבלאות מוכנות, ניתן לחזור לעבודה. חזרו לדף orderproc.asp, והקלידו בהמשך:

```
sql="INSERT INTO orders ( order_date, visitor_id,"
sql=sql & " visitor_name, email, address, phone, "
sql=sql & " credit_card_id, credit_card_num, "
sql=sql & " credit_card_exp_date, total )"
sql=sql & " SELECT Now() AS Expr1, "
sql=sql & session("visitor_id") & " AS Expr2, '"
sql=sql & visitor_name & "' AS Expr3, '"
sql=sql & email & "' AS Expr4, '"
sql=sql & address & "' AS Expr5, '"
sql=sql & phone & "' AS Expr6, "
sql=sql & credit_card_id & " AS Expr10, "
sql=sql & credit_card_num & " AS Expr9, "
sql=sql & "#" & credit_card_exp_month & "/01/"
sql=sql & credit_card_exp_year & "#" & " AS Expr8, "
sql=sql & grandtotal & " AS Expr7"

conn.Execute sql
```

הסבר: בנינו לתוך טבלת orders פקודת insert פשוטה. הערכים שהכנסנו לטבלה מוכרים מאוד, אלה בדיוק אותם ערכים שהכנסנו לטבלת visitors, אך בהבדל אחד קטן: כאן יש גם סכום לתשלום. שוב אותה טכניקה של בניית שאילתה ארוכה בשורות רבות. זה מקל על הבנייה וגם על איתור תקלות.

בסיום הבנייה, כרגיל, הרצת השאילתה כנגד הקישור הפתוח כעת.

משימה חמישית - מציאת מספר ההזמנה

כעת היינו רוצים להתחיל לבנות את שורות ההזמנה, אך יש בעיה קטנה. הנתון החשוב ביותר בכל שורת הזמנה, הוא מספר ההזמנה. אותו מספר מקשר, שבלעדיו אין לשורה משמעות. המספר נוצר אוטומטית בעת הכנסת השורה לטבלה, ואנו חייבים למצוא אותו, לפני שנוכל להכניס שורות הזמנה. נעשה זאת בעזרת שאילתה. הקלידו:

```
dim order_id
sql="select top 1 order_id from orders where visitor_id=" & session("visitor_id") &_
" order by order_id desc"
set rs=Server.CreateObject("ADODB.recordset")
rs.Open sql, conn
order_id=rs("order_id")
rs.Close
```

הסבר: המצאנו משתנה חדש order_id, שיקבל את המספר המבוקש.

כתבנו שאילתה, המוצאת את מספר ההזמנה הגדול ביותר שיש לגולש זה. זו חייבת להיות ההזמנה האחרונה.

פתחנו את הרשימה עם השאילתה המופיעה למעלה, וקיבלנו את מספר ההזמנה. סגרנו את הרשימה.



הערה:

הפקודה TOP אינה חלק מהתקן ANSI SQL 92. היא קיימת ב-Access וב- SQL Server. במסדי נתונים אחרים, אולי תידרשו לשיטות אחרות למציאת ההזמנה האחרונה, למשל,

select max(order_id) from orders…

משימה שישית - הוספת שורות הזמנה

כעת, כשמספר ההזמנה בידנו, ניתן להוסיף שורות להזמנה. יש לעבור על כל רשימת הספרים בסל הקניות, ועבור כל אחד מהם להוסיף שורה להזמנה:

```
sql="select * from basket_list where visitor_id=" & session("visitor_id")
   rs.Open sql, conn
   while not rs.EOF
     sql="insert into order_lines (order_id, book_id, title, price, quantity, total)"
     sql=sql & " select " & order_id & " as a, "
     sql=sql & rs("book_id") & " as b, '"
     sql=sql & rs("title") & "' as c, "
     sql=sql & rs("price") & " as d, "
     sql=sql & rs("quantity") & " as e, "
     sql=sql & rs("total") & " as f"

     conn.Execute sql
     rs.MoveNext
   wend
```

הסבר: יצרנו שאילתה, המביאה את כל סל הקניות של הגולש הנוכחי. שימו לב לכך שמותר לכתוב: Select * from basket_list, למרות ש- basket_list אינה טבלה אלא שאילתה.

עבור כל שורה ברשימה שהתקבלה, יוצרים שאילתת הוספה – Insert. השאילתה מוסיפה שורה לטבלת שורות הזמנה – order_lines. כל שורה מקבלת את מספר ההזמנה, ואת כל פרטי הספר כפי שהם רשומים בסל הקניות.

מפעילים את שאילתת ההוספה ועוברים לספר הבא, עד שמגיעים לספר האחרון.

פרק 15

משלוח דואר

טיפול בדואר אלקטרוני הוא חלק חשוב מפעילותו של כל אתר פעיל. בחנות הספרים שלנו אנו רוצים להודות לגולש היקר על שקנה אצלנו, ולתת לו אישור על קנייתו. במקרים אחרים, השימושים יהיו שונים. למשל: משלוח דואר עקב מילוי פרטים או הורדת תוכנה, משלוח לרשימת תפוצה באופן קבוע וכן הלאה. הסיבות שונות, אבל הצורך הוא ודאי: על האתר שלנו להיות מסוגל לשלוח (ולקבל!) דואר אלקטרוני בעצמו, באופן אוטומטי. בכך נטפל בפרק זה. נשתמש בתיקיה CDONTS, וזה אומר, שפרק זה רלוונטי רק למשתמשי NT. למשתמשי PWS, כל האמור כאן הוא בגדר השכלה כללית, ולא יעבוד אצלכם.

השירות SMTP

Simple Mail Transfer Protocol, הוא שירות (Service), במערכת הפעלה NT. התקנתו היא אחת האפשרויות בעת התקנת IIS. יש לסמן את האפשרות SMTP Service. כיצד נבדוק אם בשרת שלנו מותקן SMTP? נקרא לו, ונראה אם יענה. לחצו Start > Run, והקלידו בחלון:

telnet <ServerName> 25



את ServerName יש כמובן להחליף בשם השרת שלכם.

25 - הוא מספר ה-port לו מאזין SMTP.

אם יש עם מי לדבר, ייפתח חלון Telnet, ותקבלו תשובה דומה לזו:

```
220-MyServer Microsoft SMTP MAIL ready at
Sun, 21 May 2000 10:12:21 - 0500 Version 5.5.1877.977.9
220 WSMTP spoken here
```

אם לא – כנראה לא מותקן SMTP בשרת שלכם. כדי להתקינו, יש להריץ את תוכנית ההתקנה של NT Option Pack, ולסמן את האפשרות המתאימה.

CDONTS

הספריה **CDONTS** – Collaboration Data Objects for NT Server, המגיעה גם היא כחלק מהחבילה Windows NT Option Pack, מספקת אובייקטים נוחים מאוד למשלוח וקבלת מסרים ב-SMTP. כדאי לדעת שישנה עוד ספריה כזו, שנקראת CDO 1.2, והיא עשירה הרבה יותר, ומכילה יותר אפשרויות (יומנים משותפים, פגישות, לוח שנה...). ספריה זו מגיעה יחד עם Microsoft Exchange Server. לעומתה, הספריה CDONTS פשוטה יותר, וקלה יותר לתכנות. במספר פקודות מזערי ניתן לשלוח Mail.

האובייקט NewMail

הספריה **CDONTS** מכילה מספר אובייקטים, אך אנו נתרכז רק באובייקט הפשוט ביותר: NewMail. הוא מיועד למשלוח פשוט ומהיר של דואר אלקטרוני לכתובת ידועה.

להלן רשימת המאפיינים של האובייקט NewMail:

הסבר	**Property - מאפיין**
כתובת נוספת למשלוח (Blind Carbon Copy)	Bcc
גוף ההודעה	Body
פורמט ההודעה: HTML- 0 – 1 – טקסט רגיל	BodyFormat
כתובת נוספת למשלוח (Carbon Copy)	Cc
מסלול בסיס לקבצים הנזכרים בהודעה	ContentBase
מסלול יחסי (לבסיס) לקבצים הנזכרים בהודעה	ContentLocation
כתובת השולח	From
חשיבות: 0 עד 3	Importance
שיטת המשלוח: 0=Mime 1=Text	MailFormat
נושא ההודעה	Subject
כתובת הנמען	To

הסבר	Property - מאפיין
כותרת נוספת להודעה	Value
גרסת CDONTS	Version

הסבר על כתובות הנמענים - ישנן שלוש אפשרויות להוסיף כתובת נמען:

1. To – הפשוטה ביותר, זו כתובת הנמען הראשי.
2. CC – נמען נוסף (Carbon Copy), הנמען רואה למי עוד נשלחה ההודעה.
3. BCC – נמען נוסף "עיוור" (Blind Carbon Copy), הנמען אינו רואה את יתר הכתובות. ההודעה תיראה לו כאילו נשלחה לו בלבד.

להלן רשימת השיטות של האובייקט NewMail:

הסבר	Method - שיטה
הצמד קובץ להודעה	AttachFile
הצמד כתובת אינטרנט להודעה	AttachURL
שלח!	Send
קבע Code Page להודעה	SetLocaleIDs

דוגמה למשלוח הודעה:

```
<%
set myMail=server.CreateObject("CDONTS.NewMail")
myMail.To="bill@sofcolsoft.com"
myMail.Body="Hi bill, Just testing my new site!"
myMail.From="roni@sifroni.co.il"
myMail.Subject="1,2,3 Testing"
myMail.BodyFormat=1

myMail.Send

Set myMail=nothing
%>
```

הסבר:

ראשית, יצרנו את myMail, שהוא אובייקט NewMail.

שנית, מילאנו כמה מהמאפיינים שלו בערכים: כתובת השולח, כתובת המקבל, תוכן ההודעה, הנושא וכך הלאה. אין חובה למלא את כל הפרטים האפשריים. מובן שאם לא תמלאו את כתובת הנמען, הודעתכם לא תגיע רחוק (בעצם, לא תגיע כלל).

לאחר מילוי הפרטים, בפקודה אחת פשוטה, אנו שולחים את ההודעה.

לסיום, כדי להשאיר שטח נקי, מוחקים את האובייקט.

משימה שביעית - משלוח דואר אלקטרוני

(המשך המשימות מהפרק הקודם, דף קבלת ההזמנה orderproc.asp)

נחזור לחנות הספרים שלנו, ולמטרה שהגדרנו בתחילת הפרק: אנו רוצים לשלוח דואר אלקטרוני לגולש שקנה ספר (או ספרים). להודעה זו מספר תפקידים:

1. לתת לקונה הרגשה טובה - מישהו ענה לו, יש עם מי לדבר, וזו לא סתם מכונה אטומה. המשוב לפעולתו נותן לגולש הרגשת ביטחון. הנה, הם ענו לי!
2. להודות לקונה על כך שקנה אצלנו. כל בעל מכולת עושה זאת, גם אנחנו יכולים.
3. לתת לקונה מידע על הזמנתו. מספר הזמנה יכול לעזור, תאריך צפוי להגעת המשלוח, דרך המשלוח שנבחרה וכן הלאה. בדרך כלל תכלול ההודעה גם את סיכום פרטי ההזמנה. זהו שלב נוסף בבדיקות נכונות ההזמנה. אם נפלה טעות, אולי יגלה אותה הקונה כבר עכשיו, ולא כשהמקרר יגיע אליו הביתה.

האם תרצו לשלוח הודעת טקסט פשוטה, או דף HTML? הברירה בידיכם. האם תרצו לשלוח הודעה בעברית או באנגלית? גם כאן תצטרכו להחליט. להודעה בעברית יתרונות ברורים, אך לא כל תוכנית דואר יודעת לקרוא הודעות עבריות, ולא כל גולש יודע מה לעשות עם הודעה באותיות עבריות, רק הפוכות – משמאל לימין. בדוגמה שלנו, למען הפשטות, נשלח הודעת טקסט אנגלית.

היכן הוא המקום המתאים ביותר לממש את משלוח ההודעה? נכון, בדף orderproc.asp. זהו הדף המקבל את פרטי ההזמנה, ופועל בהתאם.

בסוף הדף, ממש לפני סגירת הקישור, נוסיף את השורות הבאות:

```
if email<>"" then
  dim myMail
        set myMail=server.CreateObject("CDONTS.Newmail")
        myMail.From="roni@sifroni.co.il"
        myMail.To=email
        b="Thank you for shoping at Sifroni Bookstore." & vbcrlf
        b=b & "The best bookstore for Web Masters and Programmers" & vbcrlf
        b=b & "Your Order number is: " & order_id & vbcrlf
        b=b & "Your Total Order is: " & grandtotal & " NIS." & vbcrlf
        b=b & "Any problems regarding this order, please refer to " & vbcrlf
        b=b & "service@sifroni.co.il" & vbcrlf

        myMail.Body=b
        myMail.Subject="Sifroni Order No. " & order_id
        myMail.Send
        Set myMail=nothing
end if
```

הסבר:

תחילה, יצרנו אובייקט NewMail.

לאחר יצירת האובייקט, אנו ממלאים את תכונותיו.

ממי נשלחת ההודעה? מאיתנו. כדאי לתת כתובת E-mail, שיוכלו לחזור אליה, ושאליה יחזור הדואר במקרה של בעיה.

אל מי נשלחת ההודעה? זה ברור, אל הלקוח.

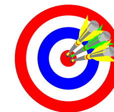
מה תהיה ההודעה? זה נתון לשיקול דעתכם. כדאי, כמו בדוגמה זו, להודיע ללקוח על מספר ההזמנה, לצורך בירורים, על סכום החיוב, על שיטת המשלוח שנבחרה ועל מועד האספקה הצפוי. אחרי כל משפט קצר כדאי להוסיף vbCrLf שפירושו: עבור שורה.

נושא ההודעה: יכול להיות שם האתר שלנו ומספר ההזמנה, כדי שיהיה בולט לעין.

לסיום, השורה החשובה ביותר: myMail.Send, גורמת לשליחת ההודעה בפועל.

אחרי המשלוח יש למחוק את האובייקט בעזרת Set myMail=nothing.

תזכורת לסיכום - כל זה יעבוד רק בשרת IIS. בשרת PWS, הספריה CDONTS לא קיימת, וממילא התוכנית לא תעבוד.



תרגול:

שכללו את משלוח הדואר. הוסיפו את פירוט הספרים שנרכשו, כמויות ומחירים.

פרק 16

החיים הקלים – Scripting Object Model

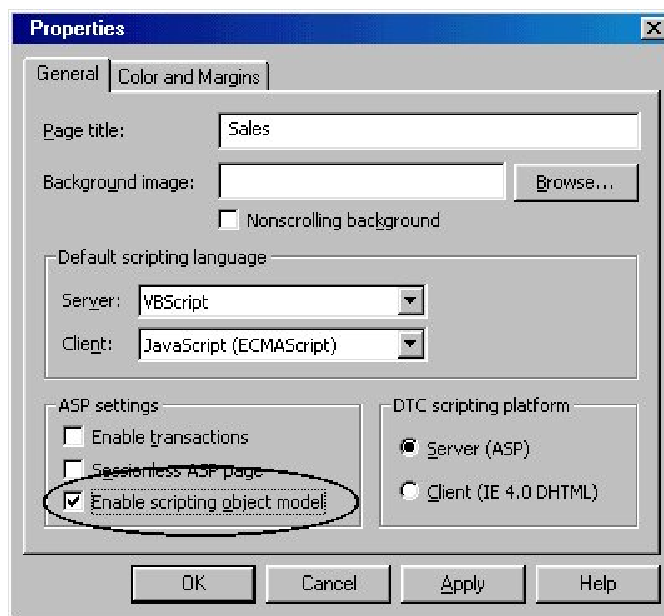
רקע כללי

Scripting Object Model, או בקיצור SOM, הוא סביבת עבודה המיועדת להקל על חיינו, ומצליחה בכך באופן מפתיע. עיקר השיפור באיכות החיים נוגע לעבודה מול מסדי נתונים, והוא מתבטא ברכיבים מוכנים המכונים Design Time Controls - DTC. עד עתה ראינו, שאפשר להקים יישומים לתפארת, גם ללא SOM. בפרק זה נלמד לנצל חלק מהיתרונות של הרכיבים המוכנים לטובתנו. כשהקמנו פרויקט חדש, חדי העין שבכם שמו לב לכך, שנוצרו כל מיני תיקיות מוזרות תחת תיקיית הפרויקט, ועד היום לא השתמשנו בהן. תיקיות אלו מכילות קטעי קוד מוכנים, המהווים "רכיבים". השימוש ברכיבים פשוט מאוד: גוררים אותם מארגז הכלים, קובעים להם כמה מאפיינים, ובדרך כלל, הם פועלים. נשמע נפלא. היתרונות ברורים: קלות, מהירות, מראה מקצועי ללא מאמץ. אך כצפוי, ישנם גם חסרונות. חסרון קל הוא נושא הביצועים. מנגנון SOM צורך משאבים רבים, ולעיתים ללא צורך. החיסרון הכבד יותר הוא בכך, שאין לנו מושג מה באמת קורה מאחורי הקלעים. אז, כשמגיעה הודעת שגיאה על שורה 3724 בפונקציה שלא אנחנו כתבנו, זה עלול להעיב על שמחתנו. רכיבי DTC הם אחד היתרונות הגדולים של פיתוח ב- Visual InterDev, וחובה להכיר אותם. עם זאת, כדאי ורצוי גם לדעת להשיג את אותן מטרות בקוד טהור, מעשה ידינו. כפי שראיתם, את רוב הפרויקט שלנו ביצענו בקוד ASP טהור, ללא שימוש ברכיבי DTC. כעת ניצור דף חדש - דף המבצעים, בעזרת רכיבים אלה.

שלב א – לאפשר את SOM

צרו דף חדש לפרויקט בשם sales.asp (לחיצה ימנית על שם הפרויקט ב- Project Explorer < Add < Active Server Page).

לפני תחילת עבודה עם SOM, יש לאפשר אותו. כלומר, להודיע לדף ASP שכעת אנו משתמשים במודל SOM. לחיצה ימנית ובחירת Properties, והחלון הבא יופיע:



נסמן את תיבת הבחירה Enable scripting object model ונאשר. נוספו כמה שורות די מפחידות לדף, אך לנו אין שום צורך או יכולת לגעת בהן. אם כבר הגענו לחלון זה, שימו לב לתיבות הבחירה של שפת ברירת המחדל. יש הפרדה בין השפה, שבה כתובות תוכניות השרת, לבין אלו שרצות במחשב הלקוח. אם שפת ברירת המחדל לשרת היא VBScript, כמו במסך, ישתול Visual InterDev בכל דף ASP חדש את השורה:

```
<%@ language=VBScript %>
```

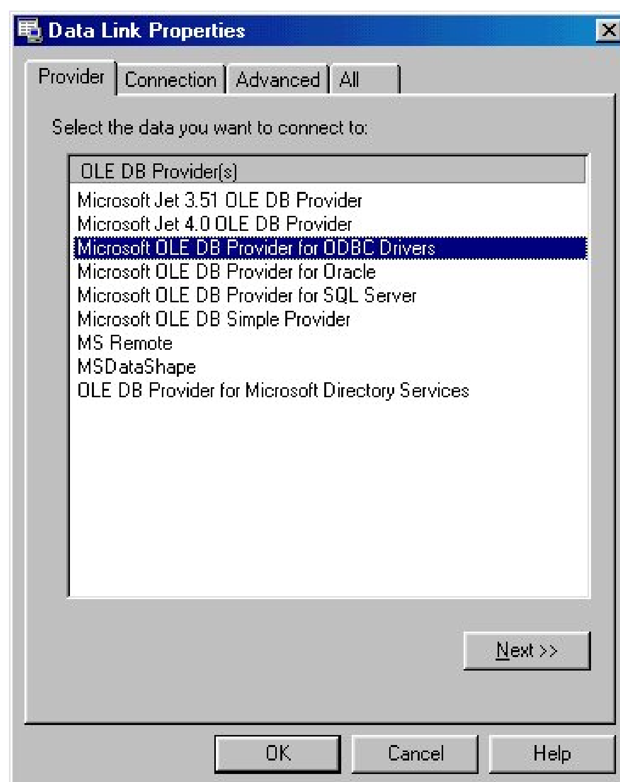
ואם השפה היא JavaScript, ייכתב:

```
<%@ language=JavaScript %>
```

שלב ב – קישור למסד הנתונים

הקדשנו זמן ואנרגיה לשיטת התקשרות מסוימת למסדי נתונים. כעת נלמד שיטה אחרת.

בחלון הפרויקט Project Explorer מצאו את הקובץ Global.asa, לחצו עליו לחיצה ימנית, ובחרו Add Data Connection. יופיע חלון דומה לזה:





הערה:

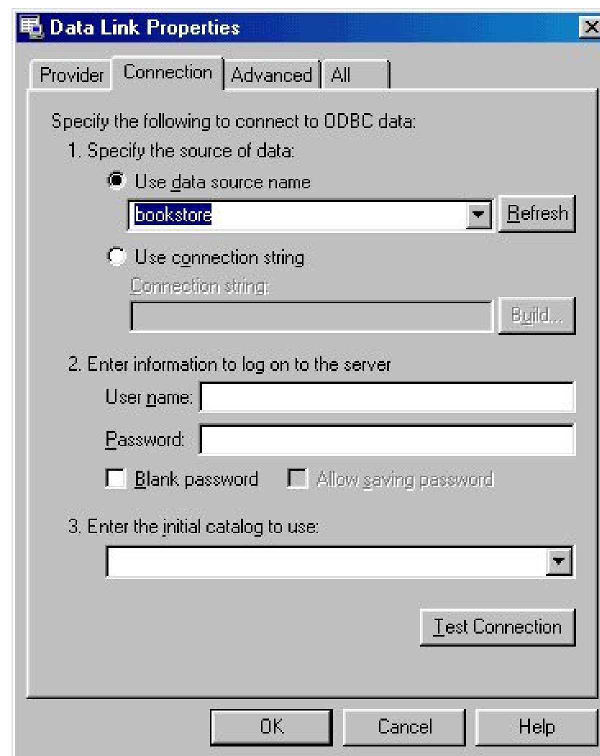
חלון זה מגיע רק אם מותקנת במחשבכם חבילת העדכון Service Pack של Visual InterDev מגירסה 3 ומעלה. אם לא – החלון נראה מעט אחרת. כדאי מאוד להתקין את חבילת העדכון, היא ניתנת להורדה חינם מאתר מיקרוסופט. גרסת ההדגמה של Visual InterDev שבתקליטור המצורף לספר, אינה כוללת את חבילת העדכון.

http://msdn.microsoft.com/vinterdev/downloads/updates.asp

החלון מחולק לארבע כרטיסיות, ובהן ניתן לקבוע את מאפייני הקישור. הכרטיסיה הראשונה שנבחר - Provider. זהו מונח השייך לטכנולוגיית OLE-DB, ופירושו – מתאם למסד נתונים ("דרייבר" בעולם ODBC). כדי שמערכת OLE-DB תדע לדבר עם מסד נתונים מסוים – חייב להיות לה מתאם - Provider אליו. כך תראו בחלון מתאמים ל-Oracle, SQL Server ואחרים.

כדי לבלבל אותנו עוד יותר, נוסף גם המתאם המשונה OLE-DB Provider For ODBC Drivers. מה זה יכול להיות? הרי בכנס מיקרוסופט אמרו לנו, ששיטת OLE-DB מחליפה את ODBC, וטובה ממנה בהרבה, ובכן, זה נכון, אבל יש מסדי נתונים שיש להם דרייבר ODBC, ואין להם עדיין מתאם OLE-DB. התקשרות OLE-DB אפשרית אל מסדי נתונים אלה דרך מתאם זה.

לדוגמה, בפרויקט שלנו, יצרנו DSN למסד הנתונים שלנו - BookStore. זהו כמובן קישור ODBC. נוכל "לדבר" עם ה-DSN שלנו באמצעות OLE-DB Provider For ODBC Drivers. בחרו בו כמסומן במסך, ולחצו Next. יופיע חלון כזה:



והשאלות הן פשוטות. אמרת ODBC? ובכן, תן לי DSN! בתיבת הבחירה ניתן לראות את כל ה-DSN במערכת, ולבחור. נבחר כמובן ב- BookStore.

סיימנו, אך לפני שנעזוב את החלון, כדאי לשים לב ליתר האפשרויות, שלא השתמשנו בהן הפעם.

אנו בחרנו DSN מוכן, מפני שיש לנו אחד. אילו לא היה לנו אחד מוכן, ניתן היה לבחור באפשרות השנייה, Use Connection String, ולבנות מכאן את הקישור למסד הנתונים.

שם משתמש וסיסמה - כפי שכבר אמרתי, גולש האינטרנט שלנו נחשב למשתמש (User) במסד הנתונים. מסדי נתונים יודעים להבחין בין משתמשים שונים, ולתת זכויות שונות למשתמשים שונים. רעיון טוב הוא, ליצור שם משתמש מיוחד לגולשים, ולשלול ממנו זכויות שונות. החוכמה היא למנוע ממנו כל אפשרות למעשים מרושעים, אך לא לפגוע בו כלל בפעילות לגיטימית. לאחר שיצרנו משתמש כזה, כאן הוא המקום לציין את שמו וסיסמתו. זהו שער הכניסה של הגולש למסד הנתונים.

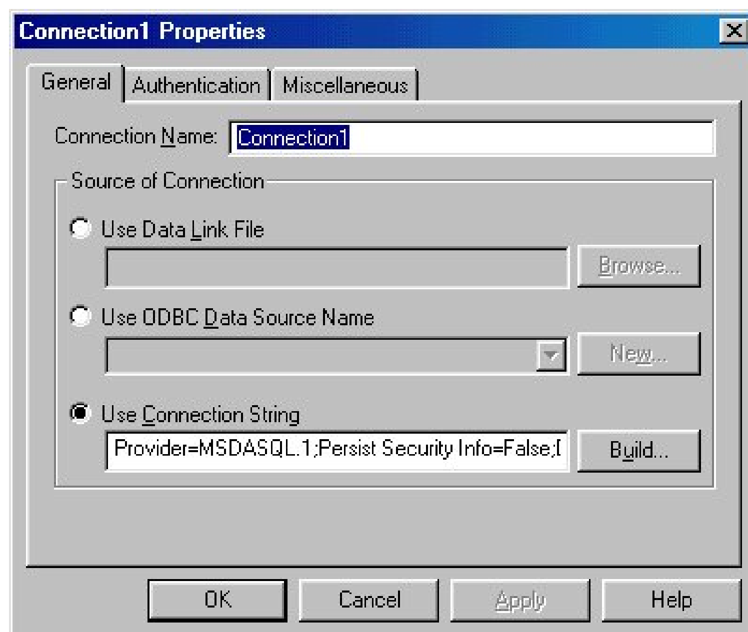
מה יקרה אם לא ציינו שם? Access, במידה ולא נקבע אחרת, מתייחס לשם ריק וסיסמה ריקה בתור המשתמש Admin, משתמש כל יכול שהכל מותר לו. זה נוח וקל למתכנתים, אך יש כאן ללא ספק פירצת אבטחה.

והבחירה האחרונה: Enter the initial catalog to use. כאשר בחרנו בקובץ Access, הגדרנו חד משמעית את מסד הנתונים מולו רוצים לעבוד. אך כאשר מתקשרים לשרת, למשל, SQL Server, יכולים להיות בו מסדי נתונים רבים. השאלה היא כמובן, מול מי אתה רוצה לעבוד. כאשר תצרו קשר עם שרת כזה, תיבת הבחירה תתמלא בשמות מסדי הנתונים המאוחסנים בו, ותוכלו לבחור.

לחצו OK, ותקבלו אישור:



יפה! יצרתם קישור. אשרו את חלון הבדיקה, והחלון הבא יופיע.



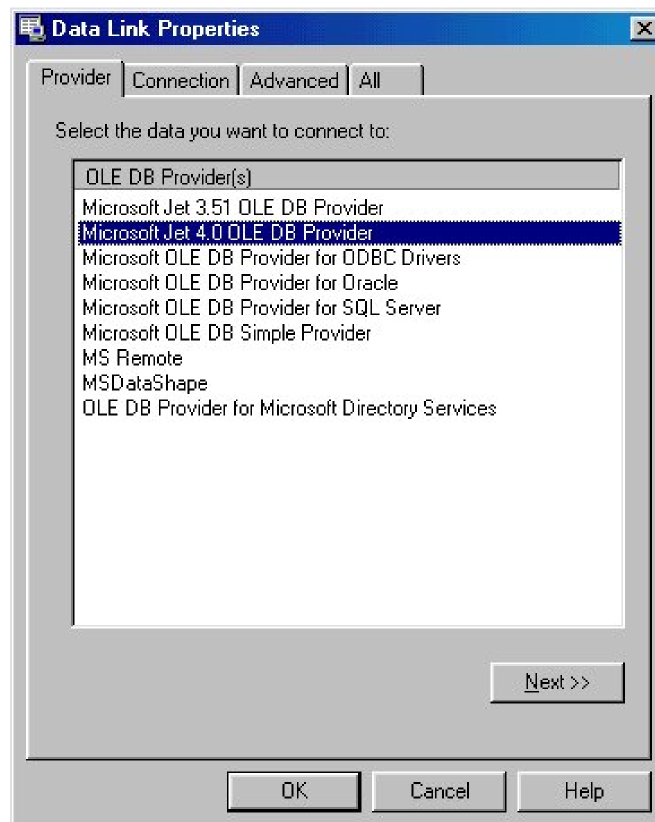
במקום השם הכללי - Connection1, כתבו Conn_ODBC, שנזכור שזהו קישור בעזרת ה- Provider של ODBC.

אבל אתם אומרים "קישור ODBC כבר היה לנו קודם, אנחנו רוצים קישור OLE-DB אמיתי!"

רוצים – תקבלו.

יצירת קישור OLE-DB

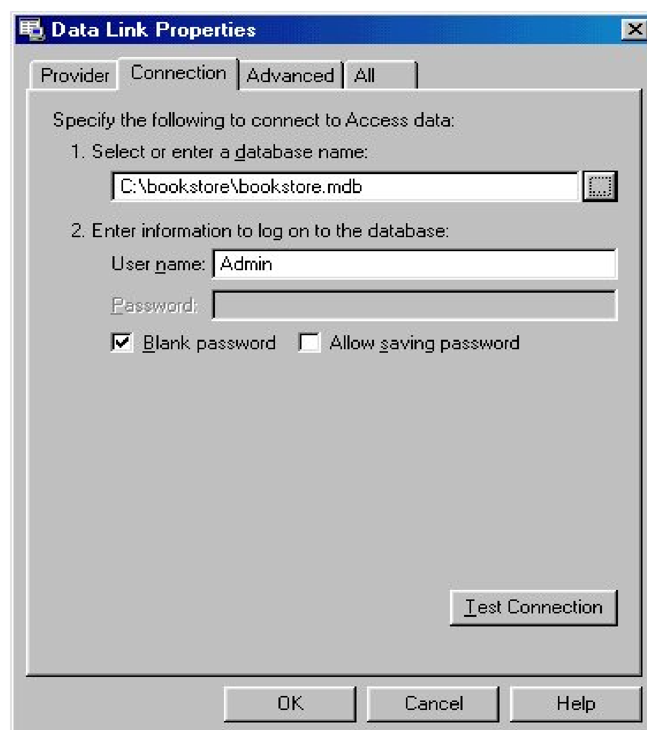
לחצו לחיצה ימנית על הקובץ Global.asa, ובחרו Add Data Connection. מופיע החלון המוכר של בחירת Provider.



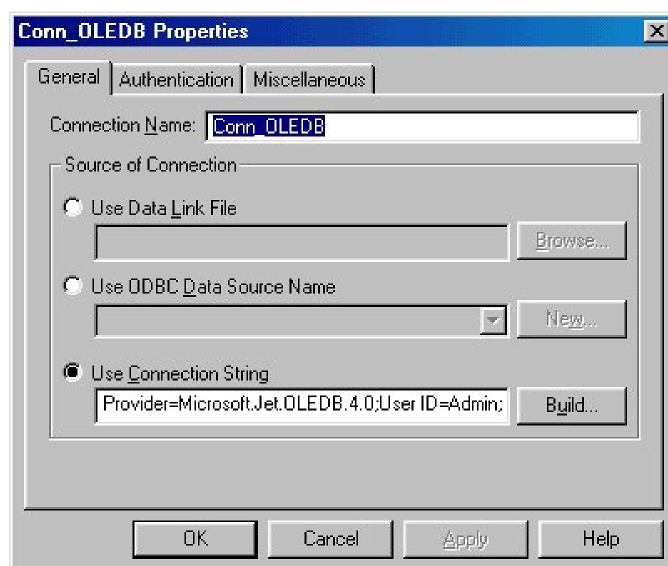
הפעם נבחר ב- Microsoft Jet 4.0 OLE DB Provider, שהוא הממשק הישיר של OLE-DB ל-Access, ונלחץ על לחצן Next.

מופיע חלון דומה מעט לחלון שהיה בדוגמה הקודמת, אך שונה. חשוב להדגיש: כל Provider מותאם למסד נתונים מסוים, ולכן הוא "שואל שאלות", שמתאימות לאותו מסד נתונים.

לדוגמה, חלון הממשק ל-Access שואל היכן נמצא הקובץ. נלחץ על הלחצן הקטן (...) ונבחר במסד הנתונים שלנו.

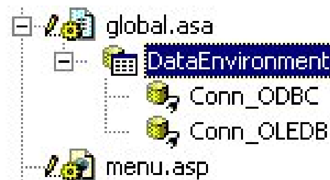


נאשר את הבחירות, ונקבל:



נשנה הפעם את השם ל- Conn_OLEDB. החלון זהה לחלון שקיבלנו בקישור הקודם, אלא שכאן שם ה-Provider שונה. אשרו את החלון.

יצרנו שני קישורים (Connections) למסד הנתונים, ואת שניהם אנו רואים בחלון הפרויקט. אנו רואים גם "יצור" חדש: Data Environment, ומתחתיו בהיררכיה את שני הקישורים שיצרנו.



Data Environment (סביבת המידע) - הוא אובייקט המייצג את כל הקשר של הפרויקט למסדי נתונים, בצורה היררכית: קישור ומתחתיו טבלאות, שאילתות וכו'. מתחת לכל טבלה/שאילתה – שדות. את כולם ניתן לראות בחלון Data Environment, ובעת הצורך גם לגרור אותם לדף ASP.



שאלה: באיזה קישור להשתמש?

תשובה: חד משמעית, השתמשו בקישור OLE-DB בכל מקום שאפשר. יש לקישור זה שני יתרונות בולטים:

1. מהירות - שיפור ביצועים עצום מושג על ידי שימוש בקישור OLE-DB.
2. תפקוד חכם ויעיל יותר - Visual InterDev מאפשר לבצע פעולות רבות על מסדי הנתונים: כתיבת שאילתות, הוספת טבלאות, Views ועוד. כל אלו אפשריים רק דרך ממשק OLE-DB.



הערה:

שימו לב לכך, שהקובץ Global.asa השתנה עקב יצירת אובייקט הקישור. למעשה, נוצר בו האובייקט בשגרה Application_OnStart.

הוספת Command

לפני הוספת Command, הבה ניזכר מה אנחנו עושים כאן: בונים את דף המבצעים המיוחדים. ישנם ספרים במבצע ומחירם, כך אנו טוענים, זול במיוחד. כדי שניתן יהיה להבחין בינם לבין ספרים רגילים, יש להוסיף שדה לטבלת books. עשו זאת.

פתחו את מסד הנתונים bookstore.mdb, בכרטיסיה tables בחרו את טבלת books, ובחרו Design.

בשורה חדשה הוסיפו שדה בשם onsale. שדה זה יסמן אם הספר במבצע או לא.

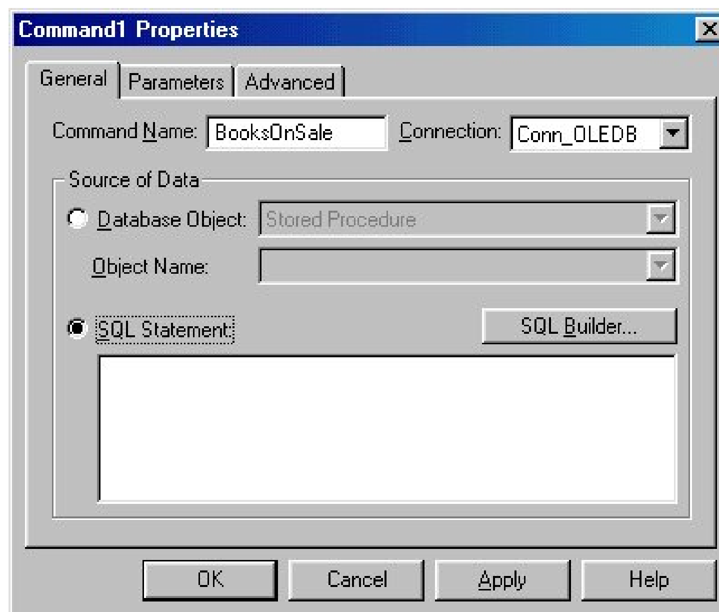
סוג המידע : Yes/No, ברירת המחדל : No.

שמרו את הטבלה.

פתחו את הטבלה ועדכנו מספר ספרים, שיהיו במבצע. כך :

onsale	review	picture
☑	ון והאבסורד.	alice1.jpg
☑	:ל על הדעת.	snark.jpg
☑	מוד והמקרא.	shirim_upizr
☑	ומד בגן גורים	gan_gurim.jp
☑	ים בגן שושנה	kippa.jpg
☑	כל גבול וחוק!	maxandmori
☐	רבים אחרים.	alice_mirror.
☐	רבים אחרים.	alice2.jpg
☐	תו של היחיד.	mea_shanim
☐	[illegible]	[illegible]

כעת אפשר ליצור Command. אך תחילה, מה זה Command? זהו אובייקט השייך למסד נתונים. הקישור מביא אותנו למסד הנתונים, אבל כאן יש אפשרויות רבות. מה ברצונך להציג? טבלה? שאילתה? שגרה? כל אחד מאלה הוא Command, ומקומו בהירככיית המידע, מתחת לקישור, שכן, הוא שייך לקישור מסוים. לדוגמה, אני יכול להתקשר לשני מסדי נתונים (כן, בו-זמנית!), באחד מהם אני משתמש בשתי טבלאות, ובשני אני משתמש בשלוש טבלאות ושאילתה אחת.



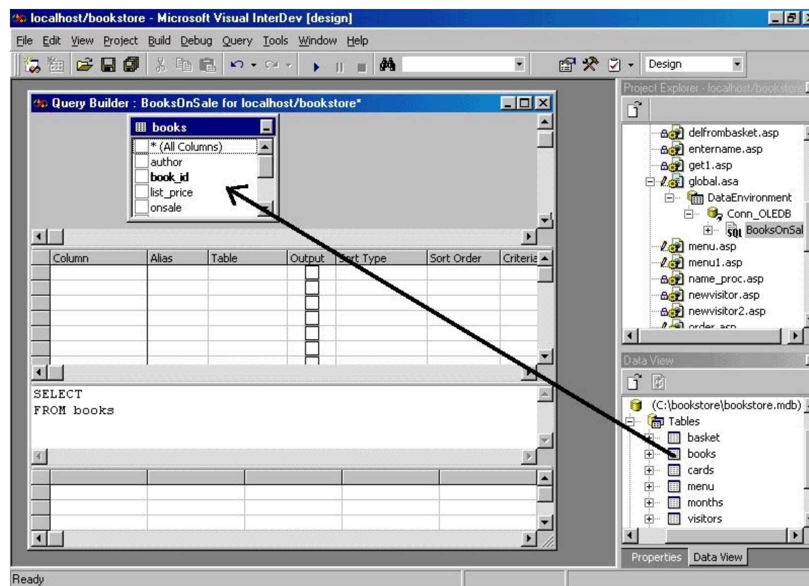
כעת לעבודה.

את הקישור בשיטת ODBC אתם יכולים למחוק, לא נשתמש בו. מעתה, כשנדבר על קישור, הכוונה היא לקישור שיצרנו בשם Conn_OLEDB.

לחצו על שם הקישור לחיצה ימנית, ובחרו Add Data Command.

בשדה השם הקלידו BooksOnSale. במקור המידע נבדוק את האפשרויות: אם נבחר אובייקט מסד נתונים, נוכל לבחור בין טבלה, שאילתה, שגרה וכו'. אם נבחר טבלה – נוכל לבחור מבין הטבלאות הקיימות. אם נבחר שאילתה – יוצגו כל השאילתות ממסד הנתונים וכן הלאה.

לשם גיוון, נבחר הפעם באופציה אחרת – שאילתת SQL. נסמן את הבחירה ואז מוצג לנו חלון ריק בו אנו מוזמנים לנסות את כוחנו ב- SQL, אבל – יש קיצור דרך. לחצו על הלחצן SQL Builder – בנאי השאילתות – ויופיע החלון הבא:



למי שמכיר את SQL Server – זוהי הפתעה נעימה. חלון כזה בדיוק יש ב- SQL Server. למי שלא – לא להיבהל, הכל כאן קל מאוד.

החלון החדש מחולק לארבעה חלונות אופקיים. נסביר כל אחד מהם:

החלון הראשון - (מלמעלה) זהו חלון הטבלאות, אליו מביאים טבלאות. הדבר הקל ביותר הוא לפתוח את חלון Data View, ולגרור משם את הטבלאות הדרושות. בואו נעשה זאת. גררו אל חלון הטבלאות את טבלת books.

החלון השני - זהו חלון פרטי השדות. כל שדה שנבחר יקבל כאן שורה, וניתן לקבוע לו פרטים שונים, כגון: שם כינוי (Alias), מיון וכו'.

החלון השלישי - זהו חלון SQL. כל פעולה שנבצע ביתר החלונות, מתורגמת כאן מייד לשאילתת SQL מדויקת. זוהי, אגב, דרך מעולה ללמוד SQL.

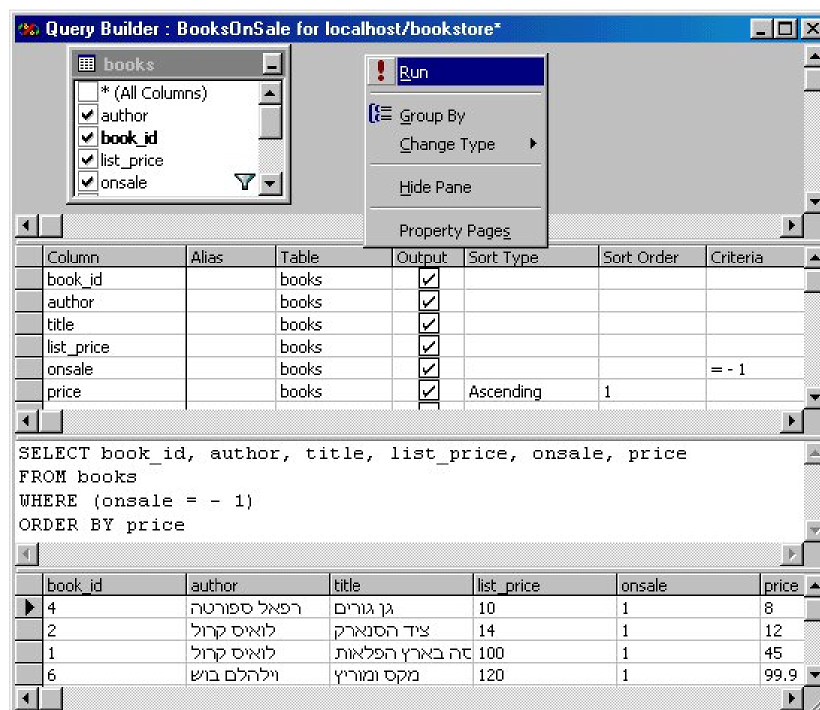
החלון הרביעי - זהו חלון התוצאה. כאשר נסיים את הגדרת השאילתה, כאן נראה את תוצאתה - הספרים שנבחרו.

כעת יש לבחור שדות. בחלון הטבלה, ליד כל שדה ישנה תיבת בחירה. סמנו את תיבות הבחירה בשדות שמעניינים אותנו : book_id ,author ,title ,list_price ,price ,onsale.

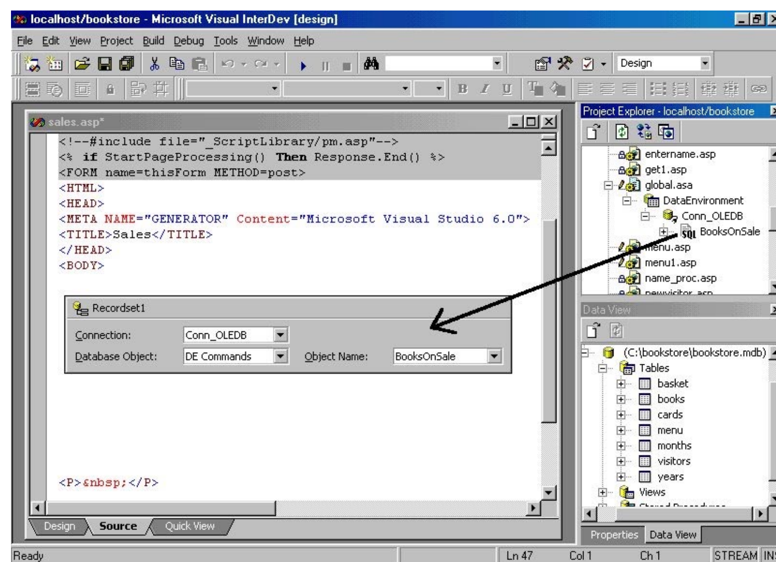
אנו מעוניינים רק בספרים שבמבצע, לכן בעמודת הקריטריון, בשדה onsale, נכתוב 1- (מינוס אחד). זהו הערך המספרי של True.

את הספרים הייתי רוצה לסדר לפי מחיר, מהזול ליקר. לכן, בסוג המיון של השדה price, נסמן Ascending – סדר עולה.

ראו כמה יפה תרגם Visual InterDev את הלחיצות שלנו לשאילתת SQL! כעת בואו נראה אם זה גם עובד. בחלון הטבלאות לחצו לחיצה ימנית ובחרו Run, כמו במסך. החלון הרביעי מתמלא בערכים. בדקו שהספרים נכונים, שאלה הם הספרים שבמבצע, ושהסדר שלהם הוא אכן מהזול ליקר.



שמרו את השאילתה, וסגרו את החלון. כעת השאילתה החדשה צריכה להופיע בחלון הפרויקט, מתחת לקישור Conn_OLEDB, הנמצא מתחת ל- Global.asa. פתחו את הדף שיצרנו - sales.asp, והביאו אליו את השאילתה החדשה. עושים זאת על ידי גרירה, כמו בתרשים שלעיל.



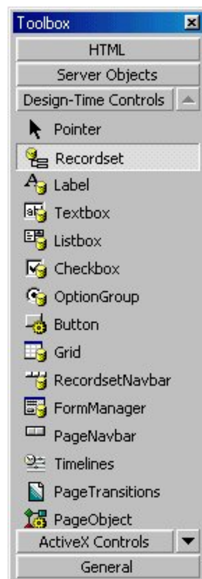
האובייקט שנוצר, הוא רכיב DTC שנקרא RecordSet. ותפקידו, בדיוק כמו ה- Recordset, שהכרנו בכתיבה רגילה – להכיל רשומות ממסד הנתונים. תוצאת שאילתה, או טבלה שלמה. דרך אחרת להשתמש בו היא לגרור אותו מארגז הכלים, ולבחור עבורו את המאפיינים הדרושים.



הערה:

הרכיב Recordset הוא רואה ואינו נראה, אין לו חלק ויזואלי. כלומר, הגולש לא יראה אותו. בזמן תכנון אנו רואים אותו, כדי שנוכל לשנות את מאפייניו.

רכיבי DTC

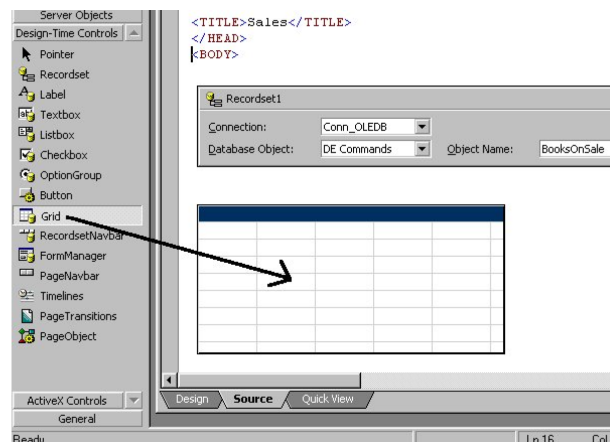


רכיבי DTC, הם רכיבי ActiveX, המאפשרים לנו לשנות את מאפייניהם בזמן התכנון, מתוך סביבת העבודה של Visual InterDev. מכאן שמם – Design Time Controls. ניתן גם לשנות את מאפייניהם בזמן ריצה, על ידי פקודות תכנות ב- VBScript או ב- JavaScript.

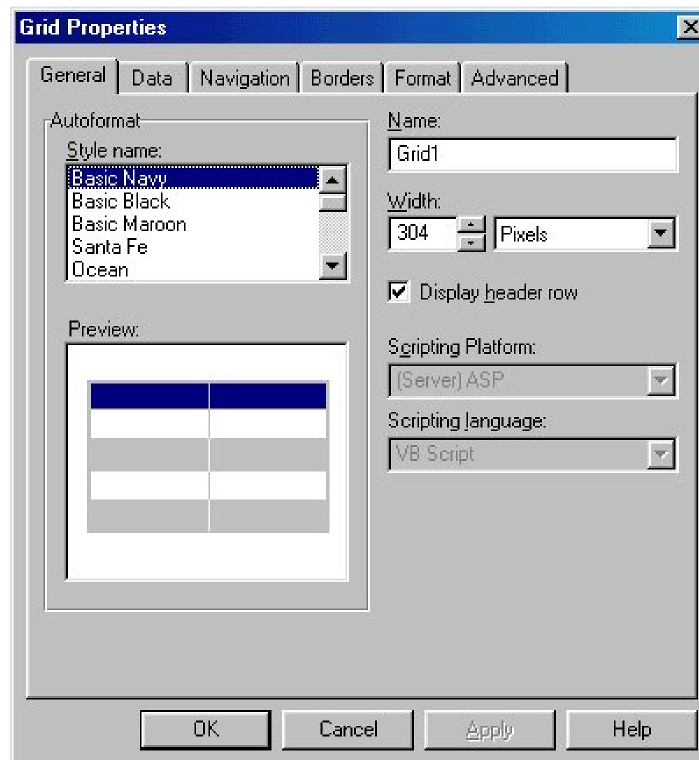
כדי לראות את רכיבי DTC, לחצו בארגז הכלים על הלחצן המסומן במילים Design Time Controls, כמו בציור.

עיקר כוחם של רכיבים אלה – בפעילות מול מסדי נתונים. הסמלים, שנקבעו עבור רכיבים אלה, דומים מאוד לרכיבים המקבילים להם ב-HTML. המבדיל ביניהם הוא סמל החבית הצהובה, סימון מקובל למסדי נתונים. תפקוד רכיבים אלה, מורכב ועשיר בהרבה מרכיבי HTML רגילים.

כעת אנו רוצים גם להציג את תוצאות השאילתה לדף. לצורך כך, נפתח את ארגז הכלים, ונבחר במחלקת Design Time Controls. משם נגרור אל הדף שלנו את הרכיב Grid. הנה כך:



כעת לחצו לחיצה ימנית על ה-Grid החדש שהבאתם, ובחרו ב- properties.



זהו חלון רב אפשרויות, כולן חשובות, ונסביר כל אחת מהן בפירוט.

General – הכרטיסיה הראשונה היא כללית. פה תקבעו מה יהיה שמו של ה-Grid החדש, מה רוחבו ובאיזו יחידת מידה. האם אתם רוצים כותרת ל-Grid? כאן תוכלו לבחור סגנון עיצובי אחד מתוך כמה סגנונות, חלקם נחמדים. בחרו סגנון ונמשיך הלאה.

Data - זוהי הכרטיסיה החשובה ביותר. כאן נקבע מה יוצג וכיצד.

Grid Properties

General | Data | Navigation | Borders | Format | Advanced

Recordset: Recordset1

Available fields: book_id, author, title, list_price

Add Unbound Column

Grid columns:

Field/Expression	Header
author	מחבר
title	title
list_price	list_price
price	price

Edit columns

Field/expression: author

Header: מחבר

Update | Delete

OK | Cancel | Apply | Help

תחילה, בחרו Recordset. זה קל, כי יש לנו רק אחד. כשבחרתם, התמלאה רשימת השדות, ואתם יכולים לבחור את מי מהם להציג. בחרו author, title, list_price או price.

בחלון האמצעי, בעזרת החיצים שבצד ימין, ניתן לקבוע את סדר העמודות. זכרו שהעמודות נכתבות משמאל לימין, לכן הסדר צריך להיות: price, אחר כך list_price, אחריו author ולבסוף title.

במסגרת Edit Columns, ניתן לקבוע את כותרות העמודות. מסמנים עמודה בחלון האמצעי, כותבים את הכותרת הרצויה בתא Header ולוחצים Update. ניתן לכתוב גם בעברית! מומלץ לתת כותרות בעברית לכל השדות.



טיפ:

בחלון Grid Properties יש באג קטן: חובה ללחוץ על Apply, לפני שעוברים לכרטיסיה אחרת. אחרת השינויים לא נשמרים!!!

Navigation - ניווט. כאן תקבעו כמה שורות מקסימום יוצגו בדף אחד, ומה יהיו הסימונים של לחצני המעבר בין דפים.

הבנתם נכון. אם למשל, השאילתה החזירה 100 ספרים, יוצגו רק 10 הראשונים, ובעזרת לחצני ניווט יוכל הגולש לעבור לדף הבא, לחזור חזרה ולשוטט כרצונו. כך נימנע מזמני הורדה בלתי סבירים. כל המנגנון הזה אוטומטי לחלוטין, ועדיין לא כתבנו שורת קוד אחת בדף הזה.

אני מציע לשנות את הערך Records/Page ל- 10, ובכל היתר לא לגעת.

לחצו על Apply, לא לשכוח.

Borders, Format – עיצובים וגבולות. כרטיסיות אלו עוסקות בענייני קוסמטיקה, ואשאיר אותן לכם לחקור לבד. רק אחזור על הטיפ: ב- Format יש 3 מחלקות: Header row המטפלת בעיצוב שורת הכותרות, All detail rows המטפלת בעיצוב שורות הגריד, ו- Specific Column הנותנת בידכם אפשרות לעצב עמודה בודדת. לאחר כל טיפול באחת מהן, לחצו על Apply. אחרת – הכל נמחק.

Advanced – מתקדם. כרטיסיה זו מאפשרת לנו לשתול תגי HTML משלנו, להשגת שליטה מירבית בתוצאה. אנו לא שוכחים לרגע, שבסופו של דבר, הגולש יראה טבלת HTML רגילה, רק שלא אנחנו כתבנו אותה. בעזרת כרטיסיה זו – אנו יכולים לעזור לה להיות מדויקת יותר. בשלב זה אין לנו מה לתרום כאן.

אשרו את החלון וסגרו אותו. אני מקווה, שקיבלתם, משהו דומה לזה:

מחיר	מחירון	מחבר	שם הספר
Xxxx [0,0]	Xxxx [0,1]	Xxxx [0,2]	Xxxx [0,3]
Xxxx [1,0]	Xxxx [1,1]	Xxxx [1,2]	Xxxx [1,3]
Xxxx [2,0]	Xxxx [2,1]	Xxxx [2,2]	Xxxx [2,3]
Xxxx [3,0]	Xxxx [3,1]	Xxxx [3,2]	Xxxx [3,3]
Xxxx [4,0]	Xxxx [4,1]	Xxxx [4,2]	Xxxx [4,3]
Xxxx [5,0]	Xxxx [5,1]	Xxxx [5,2]	Xxxx [5,3]
Xxxx [6,0]	Xxxx [6,1]	Xxxx [6,2]	Xxxx [6,3]
Xxxx [7,0]	Xxxx [7,1]	Xxxx [7,2]	Xxxx [7,3]

כעת אתם יכולים לשחק עם גודל ה-Grid כולו, ועם רוחבי העמודות. קבעו את רוחב ה-Grid למשהו סביר, ושנו את רוחב העמודות. התחילו משמאל ועברו לימין. הציבו את העכבר בין עמודת "מחירון" ל-"מחיר", וגררו את הגבול ימינה. עצבו לפי טעמכם.

חזרו למאפייני ה-Grid (לחיצה ימנית, Properties), ובצעו את השינויים הבאים:

1. יישור כותרות למרכז, נתונים ימינה.
2. מחיר – בצבע אדום, בולט (Bold).

התוצאה צריכה להיראות דומה לזה:

Sales - Microsoft Internet Explorer

Address http://localhost/bookstore/sales.asp

שם הספר	מחבר	מחירון	מחיר
גן גורים	רפאל ספורטה	10	8
ציד הסנארק	לואיס קרול	14	12
עליסה בארץ הפלאות	לואיס קרול	100	45
מקס ומוריץ	וילהלם בוש	120	99.9
שירים ופזמונות לילדים	ח. נ. ביאליק	123	110
כיפה אדומה	א. זאב	149.9	129.9

|< << >> >| 1 / 1

אני מקווה שאתם מבינים את העוצמה האדירה, את החסכון בזמן, את הגמישות והאפשרויות הבלתי מוגבלות שנותן לנו הרכיב Grid.

שימושים מתקדמים ב-Grid

כפי שראינו בדוגמה האחרונה, רכיב ה-Grid הוא פשוט המצאה גאונית. השימוש בו קל ביותר, התוצאה מרשימה ומתקבלת בזמן קצר. הבעיה היא, שלא תמיד אנו מסתפקים בתצוגה, יפה ככל שתהיה, של הנתונים. לעיתים קרובות אנו רוצים לבצע פעולות שונות על הנתונים מתוך ה-Grid. פעולות אלה אינן טריוויאליות, וכדאי להסביר אותן.

ה-Grid הוא רכיב ספריה. הוא לא צמח יש מאין. מישהו במיקרוסופט כתב אותו. הקוד, אגב, פתוח לרשותכם. תמצאו אותו בתיקיה _ScriptLibrary בפרויקט שלכם, תחת השם DataGrid. אם תעיינו בקוד, תראו, בין היתר, שהוא בשפת JavaScript, שפה שפותחה על ידי Netscape, ולא ב- VBScript, שפותחה על ידי מיקרוסופט. לעובדה זו יש השפעות מעשיות מאוד על השימוש ב-Grid. בחלון מאפייני ה-Grid בכרטיסיה Data יש שדה המאפשר לכם לשנות את Field/Expression. ניתן לכתוב בו שם של שדה פשוט, וניתן לכתוב בו גם ביטוי. זהו הפתח, דרכו אתם יכולים לחדור לעולמו של ה-Grid, ולהכתיב לו התנהגות שונה מברירת המחדל. יש כמה כללים שעליהם תצטרכו לשמור:

1. לכתוב ב- JavaScript, לא ב- VBScript. זאת מהסיבה האמורה, שכל ה-Grid כתוב ב- JavaScript.
2. שמות שדות חייבים להיות מוקפים בסוגריים מרובעים, כך: [FirstName].
3. הצמדת מחרוזות תיעשה בעזרת + ולא &.
4. לא לשכוח את סימן השוויון = בתחילת הביטוי.

5. כל קוד HTML, שברצונכם לכלול, חייב להיות בתוך מרכאות, כך: "</A>".

6. הביטוי כולו חייב להיות מחרוזת כמו בדוגמה זו:

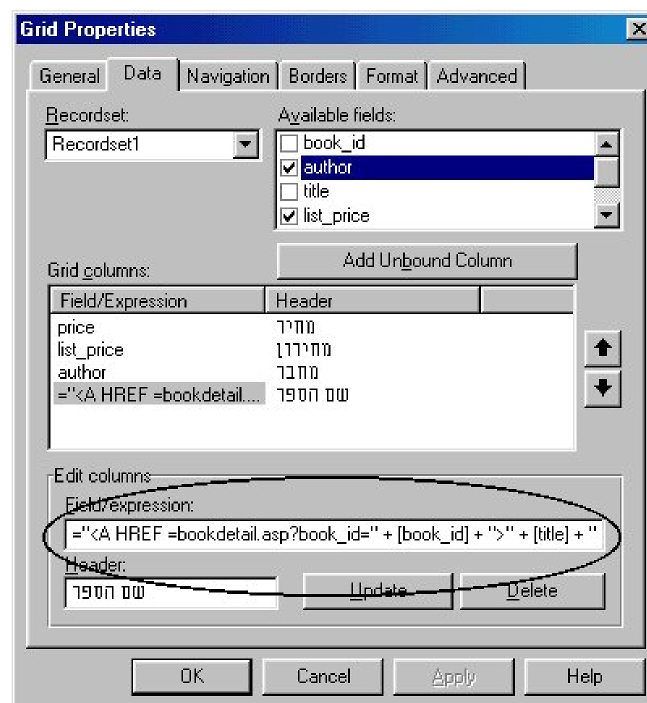
="<B><I>" + [Title] + "</B></I>"

ביצוע קישור מתוך רכיב Grid

בדוגמה הבאה נקשר את שם הספר לדף הספר הבודד - bookdetail.asp. בדיוק כמו שמתוך רשימת תוצאות החיפוש ניתן היה לקפוץ לספר בודד, אנו רוצים שגם מתוך רשימת הספרים במבצע, יוכל הגולש להגיע לפרטי ספר, לעיין בו ואם אפשר גם לקנות אותו.

לחצו לחיצה ימנית על רכיב ה-Grid, ובחרו Properties. בחלון המאפיינים בחרו בכרטיסיה Data.

בטבלה המרכזית סמנו את השדה "שם הספר". בחלק התחתון של החלון, בשדה Field/Expression בו כתוב כעת title, כתבו במקום זאת כך:



="<A HREF =bookdetail.asp?book_id=" + [book_id] + ">" + [title] + "</A>"

ובעברית: צור קישור לדף bookdetail.asp עם פרמטר book_id, שהוא קוד הספר הנוכחי. שימו לב שהקפדתי לעבוד לפי הכללים: שווה בתחילת הביטוי, שדות בתוך סוגריים מרובעים, הצמדה בעזרת + וכל השאר.

לחצו על לחצן Update, ואחר כך על OK לסגירת החלון. הריצו את הדף ותקבלו דף דומה לזה:

שם הספר	מחבר	מחירון	מחיר
גן גורים	רפאל ספורטה	10	8
ציד הסנארק	לואיס קרול	14	12
עליסה בארץ הפלאות	לואיס קרול	100	45
מקס ומוריץ	וילהלם בוש	120	99.9
שירים ופזמונות לילדים	ח. נ. ביאליק	123	110
כיפה אדומה	א. זאב	149.9	129.9

שימו לב שהסמן שצורתו יד, מעיד על קישור לדף אחר, וכן לשורת הסטטוס המראה לאן הקישור: לדף הספר עם פרמטר קוד הספר.

בצורה דומה תוכלו לשבץ תמונה קטנה בתוך ה-Grid, לחצן, או רכיב קלט כדוגמת תיבת סימון.

מידע נוסף על רכיב ה-Grid והשימוש בו תוכלו למצוא ב- MSDN.

פרק 17

מה בתפריט?

בניית תפריט פשוט

תפריטים הם חלק בלתי נפרד מכל אתר אינטרנט. בפרק זה אסקור כמה מהאפשרויות העומדות בפנינו בנושא זה.

תפריט, כמשמעותו באתרי אינטרנט, הוא בדרך כלל, אוסף של קישורים (Links) לדפי האתר. קישורים אלה יכולים להיות טקסט פשוט, אך בדרך כלל, הם יוצגו על ידי לחצנים. אלה קבצי תמונה קטנים, המשתלבים בעיצוב הכללי של האתר, ונותנים לגולש סביבה ידידותית ונוחה יותר לעבודה. באתרים מתוחכמים הלחצנים יהיו בעלי "התנהגות תלת-מימדית". כלומר: כאשר לוחצים על הלחצן, מתחלפת התמונה לתמונה של לחצן לחוץ. כך מתקבלת אשליה של לחיצה אמיתית על חפץ תלת-מימדי. ובאתרים עוד יותר מתוחכמים, כבר הצגת העכבר מעל הלחצן (mouse over) משנה אותו, וגורמת תחושת דינמיות.

גם באתר שלנו יהיה תפריט, ולא סתם תפריט, אלא תפריט בסגנון עדות ה-ASP. נבנה אותו בשני שלבים: מתחילים ומתקדמים. כל שלב יכול לעמוד בפני עצמו. בסיום הפרק תכירו את שתי השיטות, שאינן היחידות, ותוכלו לבחור איזו מהן מתאימה לאתר שלכם.

הרעיון פשוט, וכמו כל דבר ב-ASP, הוא מבוסס על טבלה. תהיה לנו טבלה, בה כל שורה היא אחת מאפשרויות התפריט. כל שורה היא, אם תרצו, לחצן. כשנרצה להציג תפריט, נסרוק את הטבלה בשיטות שלמדנו, ונדפיס למסך כל שורה בתורה. במשתנה שנקצה במיוחד לכך, נשמור את מספר הלחצן הפעיל (איזה משתנה זה יהיה? נכון, משתנה session, שהוא פרטי לכל לקוח!). במהלך הסריקה, נבדוק לאיזה מספר בטבלה הגענו. אם הוא זהה למשתנה שלנו, סימן שזהו הלחצן הנבחר, ויש להציג אותו בהדגשה. אם לא – זהו לחצן רגיל, וכך יש להציגו.

כעת לעבודה.

פתחו את מסד הנתונים שלנו - bookstore.mdb, והוסיפו לו טבלה חדשה בשם menu.
שדות הטבלה יהיו:

שם השדה	סוג נתונים	הסבר
Menu_id	Number, long	קוד שורה. מספר שלם פשוט. זהו המפתח הראשי.
Menu_text	Text	טקסט התפריט. זהו הטקסט שיוצג למשתמש, אם עובדים בשיטה של קישורים פשוטים. כשנעבוד עם לחצנים גרפיים – נשאיר אותו לצורך התמצאות שלנו (המתכנתים) בטבלה, אך לא נשתמש בו.
Action	Text	פעולת התפריט – לאן ללכת כשהמשתמש לוחץ על לחצן זה? בדרך כלל זו תהיה הפניה לדף כלשהו.

יופי, יש טבלה. כעת אנו צריכים להחליט מה יהיו אפשרויות התפריט שלנו. הרעיון הוא רק להדגים כיצד עושים תפריט, לכן בחירת הלחצנים לא ממש חשובה. אני מציע סדר לחצנים אופקי, בראש המסך, משהו דומה לזה:

חיפוש	סל קניות	הצטרף למועדון!	מבצעים	אודות ספרוני

לחיצה על לחצן "חיפוש" תוביל אותנו לדף searchbook.asp, לחיצה על "סל הקניות" תוביל אותנו לדף showbasket.asp וכן הלאה. בואו נמלא את הטבלה שיצרנו:

menu : Table

menu_id	menu_text	action
1	חיפוש	searchbook.asp
2	סל הקניות שלך	showbasket.asp
3	הצטרף למועדון	joinclub.asp
4	מבצעים	sales.asp
5	אודות ספרוני	about.asp
0		

Record: 1 of 5

מצוין. עכשיו אפשר להתחיל לבנות את דף התפריט. הוסיפו דף חדש לפרויקט שלנו, שייקרא - menu.asp. בדף זה הקלידו את הקוד הבא (קוד המקור נמצא ב-menu1.asp) :

```
<%
dim conn2
set conn2=Server.CreateObject("ADODB.connection")
conn2.open "dsn=bookstore"

dim rs2, sql2
set rs2=Server.CreateObject("ADODB.recordset")
sql2="select * from menu order by menu_id desc"
rs2.Open sql2, conn2

Response.Write "<center><table border=3 cellpadding=5>"
Response.Write "<tr>"
do while not rs2.EOF
 Response.Write "<td width=100 align=center>"
 Response.Write "<A HREF=" & rs2("action") & ">"
 Response.Write rs2("menu_text") & "</A></td>"

 rs2.MoveNext
loop
Response.Write "</tr>"
Response.Write "</center></table>"
rs2.Close
set rs2=nothing
conn2.Close
set conn2=nothing
%>
```

הסבר :

את 7 השורות הראשונות אני מקווה שאין צורך להסביר, למי שהגיע עד כאן. פתיחת קישור ורשימה סטנדרטיים לחלוטין. כדאי לשים לב לשאילתת SQL.

המילים Order by menu_id desc פירושן : סדר את הטבלה לפי השדה menu_id **בסדר יורד**. וזאת למה? הטבלה, כזכור, נכתבת משמאל לימין. ואני רוצה דווקא את הלחצן הראשון בצד ימין. כדי להשיג זאת אני פשוט קורא את הטבלה בסדר הפוך. מותר לכם להוריד את הסיומת desc מהשאילתה, ולראות כיצד מתנהג התפריט (רמז : הוא יהיה הפוך, "חיפוש" בצד שמאל).

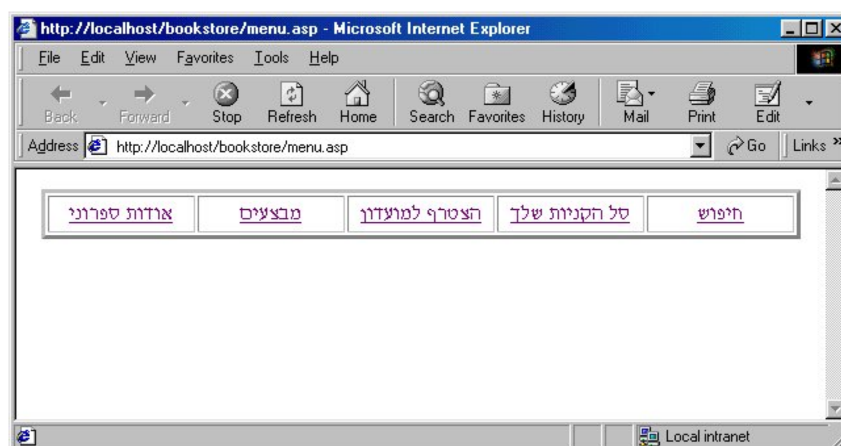
פתחתי את הרשימה ומייד אני מתחיל בכתיבת טבלה. שימו לב לכך שהפעם, שלא כתמיד, התגים <TR>, </TR> נכתבים מחוץ ללולאה של קריאת הטבלה. למה? (תחשבו לבד!).

מפני שזו טבלה עם שורה אחת בלבד! עבור כל "לחצן" אנו יוצרים עמודה בלבד, כלומר <TD>.

בטריק של הצגת קישור המבוסס על נתוני טבלה, כבר השתמשנו בעבר (היכן?), ואין בו חידוש עקרוני. המעניין כאן הוא, שהפעם שני הפרמטרים של הקישור – ההפניה והכיתוב – שניהם באים מנתוני הטבלה! עבור כל שורה בטבלת menu אנו יוצרים קישור. הכתובת אליה מצביע הקישור הוא נתון מהטבלה, השדה action. ואילו הכיתוב של הקישור, גם הוא נתון מהטבלה, זהו השדה menu_text. זה כל הפטנט.

והערה אחרונה: לכל המשתנים המוכרים שלנו הוספתי את הספרה 2. דף זה עומד להיות "משורשר" לדפי ASP הקודמים שבנינו. אסור להגדיר אותו שם משתנה פעמיים באותו דף, לכן שיניתי מעט את שמות המשתנים, כדי למנוע כפילות והודעות שגיאה.

הריצו את הדף. הוא צריך להיראות בדומה לזה:



נחמד. ראינו כבר תפריטים מיוחדים יותר, אבל זה שלנו וזה עובד. לחצו על הלחצן "חיפוש" ותגיעו מייד לדף החיפוש. אם כתבתם נכון, תגיעו לדף החיפוש. ומה קרה? התפריט נעלם!

ברור שהוא נעלם. הוא היה דף ASP עצמאי, ואנו עברנו לדף אחר! יש למצוא דרך לשלב את התפריט החדש שלנו בכל אחד מהדפים שכתבנו. ובעברית: Server Side Includes.

Server Side Includes

Server Side Includes, היא שיטה להכללת דף אחד בתוך דף אחר. בדיוק כמו במקרה שלנו. אנו רוצים לשתול את דף התפריט בכל אחד מהדפים שכתבנו. יכולנו להעתיק את קוד כתיבת התפריט לכל אחד מהדפים שלנו, בחלקו העליון, וכך היה תפריט לכולם. אבל:

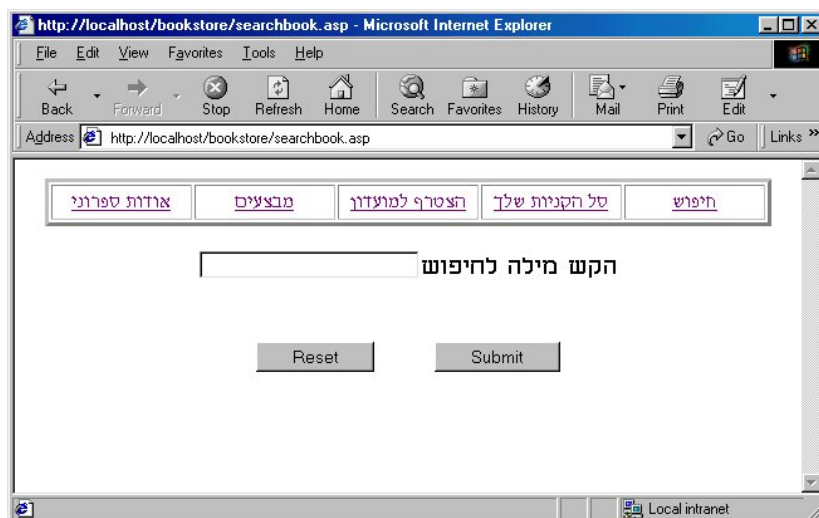
1. אנחנו עצלנים.
2. אנחנו רוצים לשפר את התפריט שלנו. בשיטת ההעתקות, נצטרך עבור כל שינוי לעדכן אותו בכל הדפים, וזו בעיה (ראה סעיף 1).

היכנסו לדף searchbook.asp, והקלידו לתוכו, מייד לאחר התג <body>, את השורה הבאה:

```
<!-- #include file = "menu.asp" -->
```

אם התחביר נראה לכם מוכר, אתם צודקים, כך כותבים הערה ב-HTML! שימו לב לכך, שהשורה חייבת להיות ב-HTML ולא ב-ASP! בדף זה לא היתה לנו כל בעיה, מפני שרובו HTML, אבל בדפים אחרים יש לזכור לכתוב את שורת ה-include מחוץ ל- <%.....%>. במקרה של תפריט, פשוט כותבים את שורת ה-include לפני ה- <% הראשון וכך פותרים את הבעיה.

מבחינת HTML זוהי הערה לכל דבר. היא גם נכתבת בצבע אפור דהוי. אבל שרת האינטרנט יידע לפענח אותה, ולהביא במקומה את דף התפריט שלנו. הריצו את הדף searchbook.asp לאחר התוספת. הוא צריך להיראות כך:



כעת התפריט הוא חלק בלתי נפרד מהדף, ואם תקישו על הלחצן "חיפוש" – תקבלו שוב את אותו דף, עם התפריט.

אם עובד לכם יפה, העתיקו את שורת include לכל הדפים בעלי החלק הוויזואלי. אני מזכיר: השורה חייבת להיות ב-HTML!

רוב הדפים יעבדו היטב, אך דף booklist.asp לא יעבוד. חפשו ספר שלא קיים ברשימה, ותקבלו את הודעת השגיאה המוכרת:

The HTTP headers are already written to the client browser.
Any HTTP header modifications must be made before writing page content.

ובכן, זו הודעה מוכרת ולא חביבה כלל, הנגרמת עקב הפעלת הפקודה Response.Redirect, לאחר פעולת שליחה כלשהי (ראו נספח שגיאות בסוף הספר). התפריט שלנו נשלח כמובן ללקוח, ולכן אי אפשר להפעיל אחריו את Redirect. מה עושים? שותלים את התפריט אחרי הבדיקה. בדף booklist.asp בודקים, כזכור לכם, שני דברים - בדיקה אחת: אם הרשימה ריקה או לא, אם ריקה – קוראים לדף אחר. בדיקה שנייה: מה כמות הספרים ברשימה? אם הכמות היא 1, קוראים ישירות לדף פרטי הספר הבודד. אין טעם להראות רשימה של שורה אחת. לבדיקות אין שום חלק ויזואלי, והן יכולות להתבצע **לפני שליחת התפריט!** אם עוברים לדף אחר, ממילא אין טעם להציג את התפריט של הדף הנוכחי. בקיצור, שתלו את שורת include לאחר הבדיקות. אני מזכיר בפעם השלישית, השורה חייבת להיות ב-HTML. ובמקרה זה, אנחנו באמצע קטע ASP. מה עושים? קוטעים אותו ב- %>, כותבים שורת include, וממשיכים אותו ב- <%. לדוגמה:

```
%>
<!-- #include file = "menu.asp" -->
<%
```

גם בדף ShowBasket.asp קיימת אותה בעיה. הפעם, התמודדו איתה לבד.

עוד שיפור אחרון לתפריט שלנו: הייתי רוצה להראות לגולש באיזה דף הוא נמצא, ובאותה הזדמנות לתת לו תחושה אמיתית ככל שניתן, של לחצן. לחצת – ומשהו השתנה. נעשה זאת בעזרת שינוי צבע הרקע של עמודת הטבלה הנבחרת. ראשית נגדיר משתנה שישמור את מספר אפשרות התפריט שנבחרה. איזה סוג משתנה זה יהיה? כמובן, משתנה session, מפני שלכל גולש יש אחד כזה, וכל אחד יכול להיות בדף לגמרי אחר. את המשתנה הזה, נזין בערך מספרי לפני הקריאה לדף התפריט. בדף התפריט נבדוק כל שורה אם היא ה"נכונה". אם כן, נצבע בוורוד כהה, אם לא – נצבע בוורוד בהיר. אפשרי גם צבעים אחרים. התוצאה הרצויה – כל התפריט ורוד בהיר, והעמודה הנבחרת – בוורוד כהה. בהזדמנות זו גם נדגיש את האותיות בעזרת <B>.

להלן השינוי הנדרש בדף menu.asp :

```
do while not rs2.EOF
   Response.Write "<td width=100 align=center "
   if rs2("menu_id")=session("menu_id") then
    Response.Write "bgcolor=DeepPink>"
   else
    Response.Write "bgcolor=Pink>"
   end if
   Response.Write "<A HREF=" & rs2("action") & ">"
   Response.Write "<b>" & rs2("menu_text") & "</b></A></td>"
  rs2.MoveNext
loop
```



הערה:
בשורה השנייה, בין המילה center לבין המרכאות יש רווח, לא לשכוח אותו!

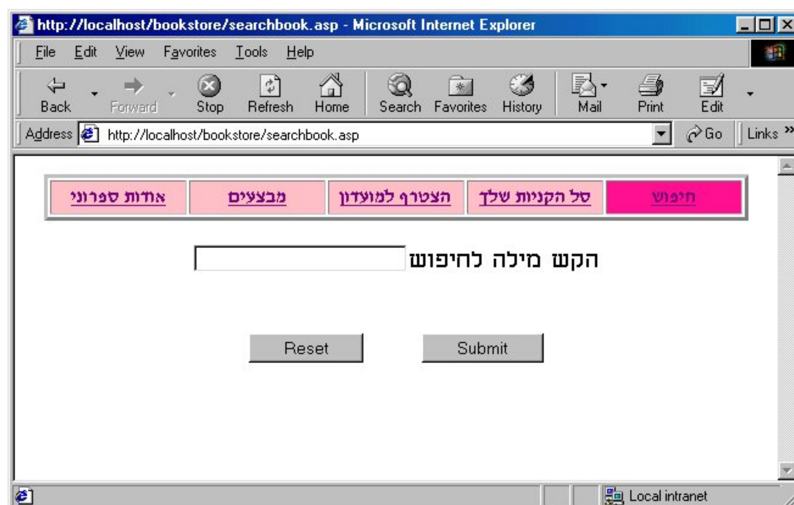
כעת, בכל דף בו מופיעה שורת ה- include, יש להוסיף מייד לפניה שורה הקובעת איזו אפשרות תפריט תידלק כאשר מוצג דף זה. קביעת השמה שכזאת היא פקודת ASP, ולכן היא חייבת להיות "עטופה" בחיצים ואחוזים כמקובל. משהו כזה :

```
<%session("menu_id")=1%>
<!-- #include file = "menu.asp" -->
```

בדפים bookdetail, booklist, searchbook, קבעו את הערך ל-1 (חיפוש).

בדפים BasketIsEmpty, ShowBasket, קבעו את הערך ל-2 (סל קניות) וכן הלאה.

כעת כאשר הגולש עובר דף, הוא מקבל על כך משוב מיידי, והוא גם יודע היכן הוא נמצא. כך בערך זה צריך להיראות:



עד כאן תפריט למתחילים. תודו שהיה קל להפליא, לא עלה הרבה כסף ושיפר בהרבה את תפקוד האתר שלנו. רוצו לספר לחברה, ולהתראות בפרק הבא – תפריט למתקדמים.

פרק 18

תפריט למתקדמים

בפרק זה נעסוק בבניית תפריט המורכב מלחצנים. כל לחצן יהיה תמונה נפרדת. ליתר דיוק, כל לחצן יהיה שתי תמונות. האחת (נקרא לה UP), היא הלחצן במצבו הלא לחוץ, והשנייה היא הלחצן כשהוא לחוץ (נקרא לה DN). הכנתי מספר לחצנים, שלכל אחד שתי גרסאות: כחול ואדום. הכחול יהיה הלחצן הבלתי לחוץ, והאדום יהיה הלחוץ. התפריט, בסופו של דבר ייראה כך:



כך משמש התפריט גם כאמצעי ניווט, וגם כמראה מקום, המסייע לגולש להתמצא, לדעת היכן הוא נמצא.

כל הלחצנים נמצאים בתקליטור המצורף לספר זה בתיקיה bookstore\buttons. נמצאים גם כמה לחצנים ללא כיתוב, כדי שתוכלו להכין תפריטים חדשים משלכם ללא מאמץ. כדי לראות את כל הלחצנים יחד, כדאי להשתמש בתוכנה נפלאה שנקראת ACDC (שנמצאת בתקליטור), זו לא תוכנה לעיבוד תמונה, אלא רק להצגה. אבל היא יודעת להציג מיגוון רחב של פורמטים, מהירה מאוד, ועובדת כמו Explorer. אתה מטייל בדיסק הקשיח ורואה את כל התמונות שנמצאות בתיקיה הנוכחית. חובה לכל בונה אתרים.

תחילה, נשכלל את הטבלה שלנו, ונוסיף לה שני שדות: button_up, button_dn. שדות אלה יכילו את שמות קבצי התמונה של הלחצנים, ממש כמו שדה התמונה בטבלת הספרים.

הזינו לטבלה את שמות הלחצנים הבאים:

menu : Table

button_dn	button_up	action	menu_text	menu_id
search_dn	search_up	searchbook.asp	חיפוש	1
basket_dn	basket_up	showbasket.asp	סל הקניות שלך	2
club_dn	club_up	joinclub.asp	הצטרף למועדון	3
sales_dn	sales_up	sales.asp	מבצעים	4
about_dn	about_up	about.asp	אודות ספרוני	5
				0

Record: 1 of 5

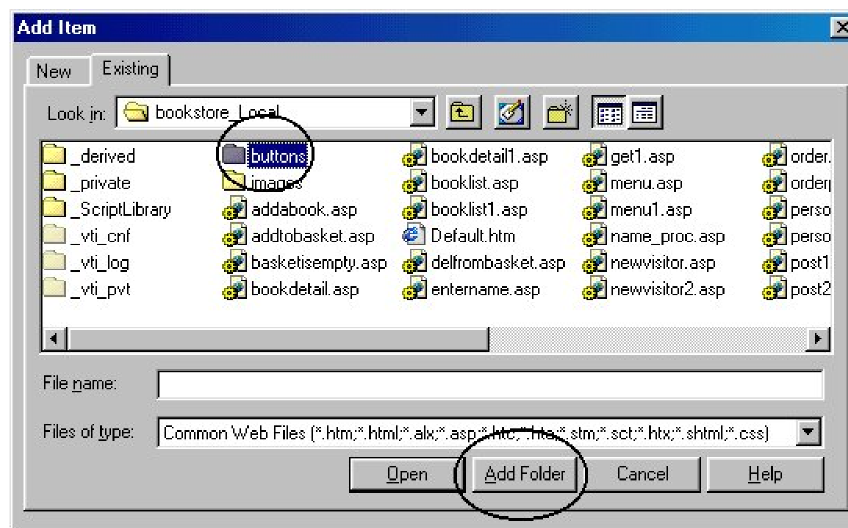
שימו לב לכך, שלא כתבנו את סיומת הקובץ. מכיון שכל הסיומות זהות – נכתוב אותן פעם אחת בתוכנית.

כעת מופיעים שמות הלחצנים בטבלה. יהיה נחמד אם הם גם יופיעו פיסית בדיסק שלכם. בעזרת סייר החלונות צרו תיקיה חדשה בתוך הפרויקט שלכם בשם buttons. התיקיה צריכה להיות בתוך התיקיה bookstore_Local, אבל לא ממש חייב. העתיקו את כל הלחצנים מהתיקיה buttons בתקליטור אל התיקיה שיצרתם.

אך זה לא מספיק. העובדה שהקבצים נמצאים פיסית בתיקיה מתחת לפרויקט, לא אומר שהפרויקט מכיר בתיקיה זו. בתוך Visual InterDev הביטו ב- Project Explorer, אין זכר לתיקיית הלחצנים. יש בכך שני דברים שליליים:

1. התוכנית לא תעבוד, היא פשוט לא תמצא את הלחצנים בתיקיה buttons, שמבחינתה לא קיים.
2. גרוע מזה, הלחצנים לא יועתקו לשרת. מנגנון העקיבה אחרי שינויים פועל רק על קבצים המשויכים לפרויקט.

לכן, חובה לצרף את תיקיית הלחצנים, על תכולתה, לפרויקט. לחצו לחיצה ימנית על שם הפרויקט (בשורה השנייה, באות עבה, ב- Project Explorer), ובחרו Add > Add Item. בחלון שנפתח, בחרו בכרטיסיה Existing. טיילו בדיסק ומצאו את התיקיה שיצרתם.



סמנו את התיקיה, כמו בתמונה, ולחצו על לחצן Add Folder. התיקיה והלחצנים שבתוכו יתווספו לעץ הפרויקט.

כעת אפשר לכתוב את הקוד. היכנסו לדף menu.asp. אתם יכולים לשמור אותו בשם אחר, למשמרת, כי אנחנו עומדים לעשות בו שינויים משמעותיים. הקפידו להמשיך בעבודה בדף menu.asp ולא בדף ששמרתם, מפני שבכל הדפים שלנו מוזכר השם menu.asp. הרעיון, בגדול, לא השתנה. סורקים את טבלת menu, ומשווים את הערך התורן של menu_id למשתנה ה- session שלנו ("Session("menu_id). אם שונה, מראים לחצן כחול. אם זהה – מראים לחצן אדום. הקוד "תפריט למתקדמים" מופיע להלן. מספר הסברים לקוד:

1. האובייקטים נקראים sql2, rs2, conn2, כדי למנוע הודעות שגיאה על הכרזה כפולה של משתנים. לא לשכוח, קוד זה "יישתל" כחלק מכל אחד מהדפים שלנו!
2. הסימן &_ משמש לשבירת שורה. פירושו: המשך המחרוזת בשורה הבאה. אצלכם זו יכולה להיות שורה אחת, ואז מספיק רק &.
3. שימו לב לשורה המודגשת. אנו סורקים את טבלת menu, והשורה בודקת: האם כבר הגענו ללחצן של הדף הנוכחי? אם לא - להציג לחצן כחול, היושב אצלנו בטבלה בשדה button_up. אם כן - יש להציג לחצן אדום, בטבלה: button_dn.
4. בין אם הלחצן כחול או אדום, אנו בונים קישור (link). הקישור הוא אל שם הדף המופיע בטבלה, בשדה action.

5. האובייקט המוצג בקישור הוא מטיפוס image ותמונתו נלקחת, איך לא, מהטבלה. המשפט הבא:

```
Response.Write "<IMG SRC='buttons\" & rs2("button_dn") &_
         ".jpg' border=0></A></td>"
```

מניח שקיים בפרויקט שלכם תיקיה buttons. אם אין, מסיבה כלשהי, תנו את המסלול המלא לקבצים. לדוגמה:

```
Response.Write "<IMG SRC='c:\bookstore\buttons\" & rs2("button_dn") &_
         ".jpg' border=0></A></td>"
```

אחרת לא תראו את הלחצנים.

6. שימו לב לתוספת **border=0** בתוך הקישור. היא תגרום לכך שלא יהיו מסגרות כחולות או סגולות סביב הלחצנים. צבעים אלה, המסמנים קישור, שימושיים בקישורים רגילים, אך בלחצנים ובקישורים גרפיים אחרים – הם מקלקלים את האשליה.

הקוד המלא לתפריט "מתקדמים", הקובץ menu.asp (קוד המקור נמצא ב-menu2.asp):

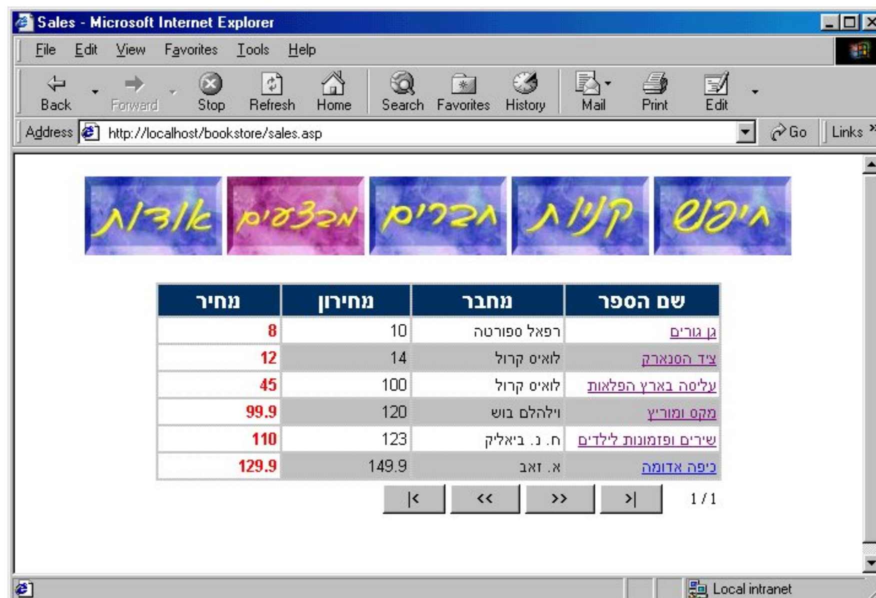
```
<%
dim conn2
set conn2=Server.CreateObject("ADODB.connection")
conn2.open "dsn=bookstore"

dim rs2, sql2
set rs2=Server.CreateObject("ADODB.recordset")
sql2="select * from menu order by menu_id desc"
rs2.Open sql2, conn2

Response.Write "<center><table>"
Response.Write "<tr>"
do while not rs2.EOF
  if rs2("menu_id")=session("menu_id") then
    Response.Write "<td><A HREF=" & rs2("action") & ">"
    Response.Write "<IMG SRC='buttons/" & rs2("button_dn") &_
     ".jpg' border=0></A></td>"
  else
    Response.Write "<td><A HREF=" & rs2("action") & ">"
    Response.Write "<IMG SRC='buttons/" & rs2("button_up") &_
     ".jpg' border=0></A></td>"
  end if
rs2.MoveNext
loop
Response.Write "</tr>"
```

```
Response.Write "</center></table>"
rs2.Close
set rs2=nothing
conn2.Close
set conn2=nothing
%>
```

כאשר תריצו אותו לבדו, תראו רק את הלחצנים הכחולים (מדוע?). כאשר תריצו כל אחד מהדפים האחרים של הפרויקט, תראו את התפריט שלכם משולב יפה בראש הדף, כאשר הלחצן הנכון מופיע בצבע אדום. תראו משהו כזה:



לחיצה על כל לחצן תביא אֶתכם לדף הנכון, וגם תצבע אותו באדום.

אם עובד לכם, זה אולי הזמן לגייס משקיעים, או לפחות להראות לאמא.

פרק 19

רכיבי Active Server

רכיבי Active server הם רכיבי תוכנה מוכנים, המיועדים לבצע מטלה מוגדרת. הם דומים מאוד לרכיבי ActiveX, בהבדל אחד חשוב: הם מותקנים ופועלים בצד השרת ולא בצד הלקוח. מכיון שרכיבים אלה עברו הידור (קומפילציה), הם מהירים בהרבה מתוכניות ASP רגילות.

למען האמת, כבר השתמשנו שימוש נרחב באחד מרכיבים אלה, הרכיב ADO. בזכותו הפכו משימות מורכבות כמו קישור למסד נתונים, לעניין של שורה בודדת.

אבל היכן משיגים רכיבים מופלאים שכאלה? ובכן, כמה מהם כבר מותקנים בשרת שלכם, כחלק מההתקנה. אחרים ניתנים להורדה חינם מהאינטרנט, ואחרים נמכרים תמורת תשלום. לאחר שתלמדו להשתמש באחד מהם, לא תהיה לכם בעיה להשתמש ברבים אחרים. ניתן כמובן גם ליצור רכיבים בעצמכם, בעזרת כלי תכנות כמו Visual Basic, או ++Visual C, אך לא נדון בכך בספר זה (תוכל למצוא על נושאים אלה בספרים "**סדנת לימוד Visual Basic 6**" ו"**סדנת לימוד ++Visual C 6**", שניהם בהוצאת הוד-עמי).

11 רכיבים מותקנים כחלק מ-ASP, והם:

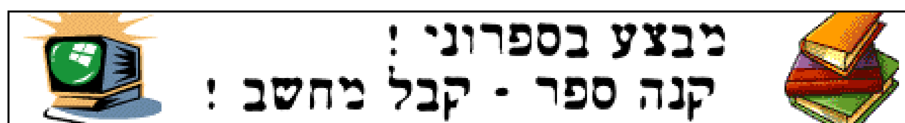
תיאור	שם הרכיב
גישה למסדי נתונים. יצירת קישור, פתיחת רשימה (Recordset), ביצוע פעולות על רשימה ועוד.	ADO
אחראי על ביצוע סבב אקראי או קבוע מראש של תמונות הפרסומת המופיעות באתר.	Ad Rotator
נותן מידע מפורט על הדפדפן הנמצא אצל הלקוח. יצרן, גירסה, מערכת הפעלה ועוד. לפי מידע זה ניתן לעצב את הדף נכון יותר.	Browser Capabilities
מאפשר ליצור דפי תוכן בקלות רבה, ומאפשר ניווט קל בין דפי האתר, בעזרת לחצני קדימה/אחורה.	Content Linking
מחליף אוטומטית את תוכן הדף לפי תבנית שנקבעת מראש.	Content Rotator
סופר אוטומטית את כמות המבקרים בכל דף, ושומר את התוצאה בקובץ ASCII.	Page Counter
בודק הרשאות. קובע אם לגולש מותרת גישה לקובץ.	Permission Checker

שם הרכיב	תיאור
Counters	כפי ששמו מעיד, מיועד ליצירת מונים. לא רק מוני מבקרים למיניהם, אלא כל מונה אחר. מונה ספרים שנרכשו, מונה הזמנות, מונה הפניות לאתר אחר ועוד.
My Info	שומר מידע אישי.
Tools	תוכניות שירות שונות, לדוגמה: FileExists, Random ועוד.
Status	מספק מידע על מצב השרת.

ונעבור לפרסומות – Ad Rotator

נושא הפרסומות באתרי אינטרנט הופך לעניין מרכזי, מפני שלעיתים קרובות הוא זה שמאפשר, כלכלית, את קיום האתר. מובן שהבעיה העיקרית היא, למצוא מי שיהיה מוכן לשלם עבור פרסומת באתר שלכם, אבל גם לאחר שמצאתם, אתם בבעיה. חתמתם הסכם עם 15 חברות לפרסם אותן באתרכם. אתם רוצים ליצור סבב פרסומות קבוע. כאשר גולש כלשהו מבקש את דף הבית – צריך להופיע בראש הבית הבאנר (Banner) הראשון ברשימה. הגולש השני, יקבל את הבאנר השני וכן הלאה. כשהרשימה מסתיימת – חוזרים להתחלה. ניתן לכתוב זאת ב-ASP, זה לא קשה במיוחד, אבל לעיתים נוח להשתמש ברכיב מוכן. הרכיב נותן ביצועים גבוהים מאוד, וגם כמה אפשרויות מתקדמות. לדוגמה: ספירת לחיצות על כל באנר (כך נדע בכמה לחייב את הלקוח), סבב אקראי או קבוע בין הבאנרים, מתן משקל יחסי (עדיפות) שונה לכל באנר ועוד.

לצורך הדוגמה שלנו, במקרה הכינותי מראש שני באנרים נחמדים:





שמות הקבצים: sif1.gif, sif2.gif. תמצאו אותם בתיקיית התמונות (bookstore\images), בתקליטור המצורף לספר. העתיקו אותם לתיקיה images בפרויקט שלכם, וצרפו אותם לפרויקט: ב- Project Explorer לחיצה ימנית על התיקיה images, בחרו Add Item, והוסיפו את התמונות.

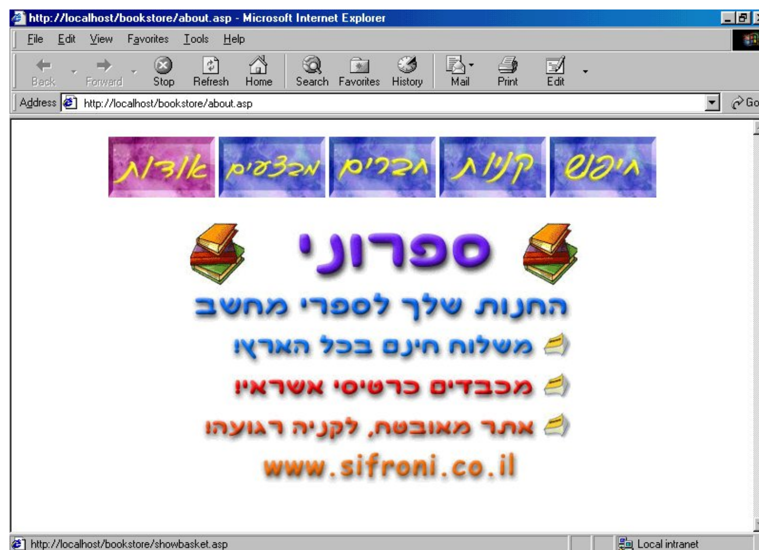
אצלנו הפרסומת תהיה עצמית. את הגולשים שילחצו על הבאנר הראשון – נפנה ישירות לדף המבצעים שלנו - sales.asp. ואת אלה שילחצו על הבאנר השני – נפנה לדף חיפוש הספרים - searchbook.asp.

נכין כעת דף פשוט, שבו יופיעו הבאנרים שלנו. הוסיפו לפרויקט דף חדש בשם - about.asp. בדף זה תהיה תמונה אחת בלבד - about.jpg, הנמצאת בתיקיית התמונות בתקליטור (bookstore\images). הוסיפו אותה לפרויקט, והוסיפו אותה לדף שיצרתם.

כעת הוסיפו שורת include בדף about.asp שתגרום לביצוע הקוד שבדף menu.asp (במקרה שלנו, יוצג התפריט). התוצאה: בדף שלנו יהיו תפריט ותמונה. הקוד צריך להיראות כך:

```
<%@ Language=VBScript %>
<html>
<head>
<meta NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</head>
<body>
<%session("menu_id")=5%>
<!-- #include file = "menu.asp" -->
<center><img SRC="images/about.jpg"></center>
</body>
</html>
```

השורה session("menu_id")=5, תגרום לתפריט להצביע על "אודות". נריץ את הדף ונקבל:



ההכנות הסתיימו, ונוכל לשוב לרכיב Ad Rotator.

מאפיינים ושיטות הרכיב

לרכיב Ad Rotator יש שיטה (method) אחת בלבד: GetAdvertisement, ותפקידה פשוט: להביא את התמונה הבאה בתור, ולהציג אותה כ-HTML.

לרכיב שלושה מאפיינים:

Border – האם יהיה גבול לאובייקט הפרסומת? 0=לא, 1=כן.

Clickable – האם האובייקט מהווה קישור?

TargetFrame – שם המסגרת בה יש להציג את האובייקט. אם עובדים במסגרות – יש לכתוב את שם המסגרת אליה יופנה הבאנר. אם לא – המאפיין תומך בכל האפשרויות הרגילות: _self, _parent, _top, _blank. האפשרות הפשוטה ביותר היא _self, העבר לדף הנוכחי.

קובץ התזמון

לפני שמתחילים לעבוד עם הרכיב Ad Rotator, יש להכין עבורו קובץ תזמון. זהו קובץ טקסט רגיל, המכיל מידע על קבצי התמונה, הגודל בו יש להציגם, וכמה זמן יש להציג כל אחד מהם. הקובץ מורכב משני חלקים, מופרדים בכוכבית ביניהם. החלק הראשון אינו חובה, ובו נקבע את מאפייני ברירת המחדל של התמונות. ניצור קובץ חדש בשם adfile.txt, ונכתוב בו כך:

```
REDIRECT /adrot.asp
WIDTH 468
HEIGHT 60
BORDER 0
*
```

השורה הראשונה חשובה מאוד: זהו הקובץ אליו תיווצר ההפניה, והוא שיבצע הפניה בפועל. כך תינתן לנו הזדמנות לבצע פעולות שונות לפני ההפניה עצמה, בעיקר פעולות של רישום בקובץ log או בטבלה כלשהי. ניתן לוותר על שורה זו, ואז תתבצע הפניה ישירה, ללא רישום. בדרך כלל חבל לוותר על אפשרות זו.

החלק השני, לאחר הכוכבית, הוא חלק חובה, והוא בעצם רשימת התמונות שיש להחליף. כל אחת עם שמה ומיקומה המדויק, וכמות הזמן היחסית שיש להציג אותה. עבור כל תמונה צריכות להיות ארבע שורות:

1. שם ומיקום התמונה.
2. כתובת שאפשר לעבור אליה, במקרה ומישהו לחץ על התמונה (כתבו מקף – אם אין קישור).
3. טקסט להצגה, אם הדפדפן אינו תומך בתמונות.

4. זמן יחסי להצגת התמונה. חשוב: זהו לא זמן ממש, אלא חלק יחסי של הגולשים, שיקבלו תמונה זו. בדוגמה שלנו, ניתן לשתי התמונות את המספר 10. לכן תהיה החלוקה ביניהן שווה. אם נקבע את המספרים 10, 20, שליש מהגולשים יקבלו תמונה אחת, ושני שליש יקבלו את התמונה השנייה.

הוסיפו לקובץ adfile.txt (לאחר הכוכבית):

```
images/sif1.gif
sales.asp
Sifroni, number 1 in computer books
10
images/sif2.gif
searchbook.asp
Sifroni, number 1 in computer books
10
```

שמרו את הקובץ וצרפו אותו לפרויקט, אחרת השרת לא יראה אותו וכל העסק לא יעבוד!!!

קובץ ההפניה – Redirection File

חשוב להדגיש: אין חובה להשתמש בקובץ הפניה. כתבנו בקובץ התזמון את היעד להפניה, וזה מספיק. אם לא נכתוב שורת REDIRECT בקובץ התזמון, הוא יפעל ללא שימוש בקובץ הפניה.

קובץ ההפניה הוא דף ASP רגיל, הנותן לנו אפשרות לבצע פעולות שונות לפני ההפניה או בנוסף אליה. פעולות שנרצה לבצע, יהיו בדרך כלל, קשורות לרישום בקבצי Log שונים, או ספירה.

השורה החשובה ביותר בקובץ ההפניה היא זו:

Response.Redirect(Request.QueryString("url"))

כלומר: קלוט את המשתנה url, והפנה את הגולש לכתובת, הנמצאת באותו משתנה.

בעת קריאה לשיטה GetAdvertisement, מחליט הרכיב Ad Rotator תורו של איזה באנר להיות מוצג. עבור באנר זה נוצר קישור לקובץ ההפניה עם פרמטר שנקרא url, המכיל את הכתובת אליה יש להפנות את הגולש. קובץ ההפניה מקבל את הפרמטר url, ויכול לעשות איתו דברים שונים, אך בסופו של דבר הוא חייב לבצע אליו הפניה.

דוגמה לקובץ הפניה, adrot.asp :

```
<%
dim url
url=Request.QueryString("url")

if instr(url,"sales") then
' Write to Advertisement history table
end if

if instr(url,"searchbook") then
' Write to Advertisement history table
end if

Response.Redirect(url)

%>
```

הסבר :

דף זה הוא רק שלד לקובץ הפניה, ובעצם מבצע רק הפניה פשוטה.

תחילה, קולטים את הפרמטר url שנשלח אלינו. לכתובת זו עלינו להפנות את הגולש בסופו של דבר. לפני ביצוע ההפניה בודקים : אם בתוך הכתובת מופיעה המילה sales, אז זו כנראה הפניה לדף המבצעים. הפונקציה instr מחפשת מחרוזת אחת בתוך מחרוזת שנייה. אם התשובה חיובית – ניתן לרשום זאת בטבלה או בקובץ Log כלשהו.

כך עוברים ובודקים את הכתובת, עבור כל אחת מההפניות האפשריות.

בסוף הדף מבצעים הפניה. ההפניה נעשית בעזרת שתי השורות הבאות :

```
url=Request.QueryString("url")
Response.Redirect(url)
```

יכולנו גם לכתוב :

```
Response.Redirect(Request.QueryString("url"))
```

וזה היה בדיוק אותו דבר.

אל תשכחו לשמור את דף ההפניה בתיקיית קבצי הפרויקט, ולשייך אותו לפרויקט, כדי שיועתק לשרת!

שימוש ברכיב

כעת, כאשר כל קבצי העזר מוכנים, ניכנס לדף about.asp, ונכתוב בראש הדף.

ליצירת מופע (Instance) של האובייקט נשתמש בשיטה המוכרת Server.CreateObject.

```
<%
dim AdRot
set AdRot=Server.CreateObject("MSWC.AdRotator")
```

יצרנו את האובייקט, כעת נקבע את תכונותיו:

```
AdRot.Border=0
AdRot.Clickable=True
AdRot.TargetFrame="_self"
```

כעת נשתמש בשיטה היחידה של האובייקט, ונקבל את הקוד להצגת התמונה:

```
mystr=adrot.GetAdvertisement("adfile.txt")
Response.Write "<center>" & mystr & "</center>" & vbcrlf
%>
```

לסיכום, דף about.asp במלואו (קוד המקור נמצא בקובץ about1.asp):

```
<%@ Language=VBScript %>
<html>
<head>
<meta NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</head>
<body>

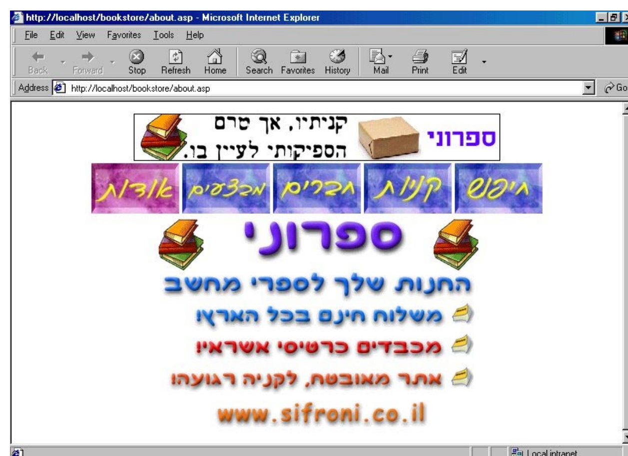
<%
dim AdRot
set AdRot =server.CreateObject("MSWC.AdRotator")
AdRot.Border=0
AdRot.Clickable=true
AdRot.TargetFrame="_self"

mystr= AdRot.GetAdvertisement("adfile.txt")
Response.Write "<center>" & mystr & "</center>" & vbcrlf

session("menu_id")=5%>
<!-- #include file = "menu.asp" -->

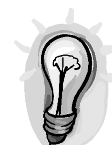
<center><img SRC="images/about.jpg"></center>
</body>
</html>
```

שמרו והריצו. התוצאה המתקבלת, צריכה להיראות בערך כך:



הבאנר מופיע בראש הדף. הציבו את הסמן על הבאנר, וראו את ההפניה בשורת הסטטוס של הדפדפן. ההפניה היא, כצפוי, לדף adrot.asp, עם פרמטר url. כעת לחצו מספר פעמים על הלחצן Refresh בדפדפן. לאחר זמן מה צריך להופיע הבאנר השני. ודאו שאכן תמונות הפרסומת "מתגלגלות" נכון. לחצו על אחד הבאנרים. אחד מהם צריך לשלוח אֶתכם לדף המבצעים, והשני לדף חיפוש הספרים. בחרו בתפריט הדפדפן View > Source, וראו את הקוד שנוצר.

לסיכום, הכרנו את רכיב Ad Rotator, אחד מהרכיבים הנקראים Active Server Components. שיחקנו מעט בפרסומות, באנרים ולחצנים. כדאי לנסות ולהשתמש גם ביתר הרכיבים ולהכיר את יכולותיהם. לעיתים הם חוסכים עבודה רבה.



טיפ:

1. כדאי להעביר את כל הקוד המטפל בבאנרים וברכיב Ad Rotator לדף התפריט - menu.asp. כך יופיעו הבאנרים בכל אחד מדפי הפרויקט ללא מאמץ.

2. רוב הבאנרים שתראו באינטרנט, מציגים אנימציה, כיתובים מתחלפים, או פעילות כלשהי. זו שיטה שנקראת Gif Animation. קובץ מסוג gif יחיד (או ליתר דיוק GIF89a), יכול להכיל תמונות רבות, עם "תוכנית עבודה" קטנה: קודם הצג תמונה זו, חכה שנייה, הצג תמונה אחרת וכך הלאה. החלפת התמונות יוצרת תחושת תנועה ודינמיות. ישנן תוכנות רבות ליצירה של Gif Animation. בדרך כלל אלה לא יהיו תוכנות ציור (כאלה יש מספיק), אלא תוכנות שמקבלות ציורים מוכנים ו"אורזות" אותם יחד לקובץ gif אחד. אחת הראשונות בתחום זה, ועדיין אחת החזקות ביותר, היא Gif Construction Set. התוכנה נמצאת בתקליטור המצורף לספר זה, וכדאי מאוד להתנסות בשימוש בה.



טיפ:

בואו נהפוך לרגע את התפקידים. נניח שאתם שותלים באנרים באתרים אחרים, כדי להגדיל את מספר הכניסות לאתר שלכם. התשלום לכל אחד מבעלי אותם אתרים מתבצע לפי מספר הכניסות, שהגיעו לאתר שלכם מהאתר שלו. עבור כל לחיצה, שגולש ביצע באתר המפרסם, על הבאנר שלכם – מגיעות לו כך וכך אגורות. אבל כיצד סופרים? כיצד נדע איזה גולש הגיע מאיזה אתר? בעיקר כאשר מפרסמים במספר גדול של אתרים.

הנה הפטנט:

שלב 1: לכל אתר מפרסם ניתן קוד. למשל, נקרא לו ad_id, קוד מפרסם.

שלב 2: בכל אחד מהבאנרים, אנו מודיעים למפרסם לאן להפנות את הגולש. לדוגמה, אם אנו רוצים להפנות לדף המבצעים שלנו, נכתוב: www.sifroni.co.il/sales.asp.

כעת, נוסיף שכלול קטן: מפרסם מספר 3 יפנה את גולשיו לכתובת: www.sifroni.co.il/sales.asp?ad_id=3.

כלומר, שולחים את הכתובת וגם פרמטר, שאומר מהיכן הגענו. ברור שבכל אתר מפרסם יהיה ad_id שונה. ניתן לשכלל את השיטה ולתת לכל דף באתר המפרסם – קוד אחר.

שלב 3: בדף שאליו מגיעים - sales.asp בדוגמה שלנו, נקלוט את הפרמטר, כך:

```
<%
dim ad_id
ad_id=Request.QueryString("ad_id")
if ad_id<>"" then
```

הוסף שורה לטבלת כניסות מאתרי מפרסמים...

```
end if
...
%>
```

נבדוק מאיזה אתר הגענו, ונוסיף שורה לטבלה מיוחדת לכך. שדות טבלה כזו יכולים להיות:

- תאריך ושעת הגעה (ברירת מחדל: ()now).
- קוד מפרסם.
- Visitor_id - בעזרתו נוכל לדעת אם התבצעה רכישה, ונדע את יעילות הפרסום, ולא רק כמה "Hits" היו.

כל חודש נספור (בעזרת שאילתה) את מספר השורות בטבלה לכל מפרסם, ונדע כמה מגיע לו.



שאלה: ואם הגולש לא הגיע דרך באנר באתר אחר, אלא גלש אלינו ביוזמתו, מה יקרה?

תשובה: שום דבר. המשתנה ad_id יהיה ריק, והוספה לטבלה לא תתבצע.

פרק 20

סיום: עולים לאוויר!

בשעה טובה ומוצלחת סיימנו את הפרויקט, והחנות הווירטואלית שלנו מוכנה לעלות לאוויר. שנייה לפני בקבוקי השמפניה, יש לטפל במספר נושאים משעממים, כמו: היכן יאוחסן האתר? כיצד נעביר לשם את הקבצים? כיצד נתחזק את האתר? והיכן יהיה מסד הנתונים?

לגבי אחסנת האתר, ישנן שתי אפשרויות עיקריות:

1. אחסון האתר בשרת מארח. משלמים סכום חודשי קבוע, ומישהו אחר דואג לתקשורת, לאבטחה, לגיבוי ולכל השאר.
2. להקים שרת אינטרנט משלנו. מובן שאז יש לשכור קו תקשורת קבוע ויציב לספק אינטרנט כלשהו.

השיטה הכדאית, בדרך כלל, היא השיטה השנייה, הקמת אתר עצמאי. קו תקשורת כזה עולה לא מעט, ויש לקוות שהמחירים יירדו עם הזמן. בשיטה זו יש לכם שליטה מירבית על הנעשה באתר. עם זאת, כדאי לדעת שמדובר בהוצאה כבדה. שרתים, רשת תקשורת, רשיונות, תוכנת FireWall, Router (נתב), יחידת אל-פסק, התקן גיבוי ועוד - לא מעט כסף. ישנן גם עלויות קבועות של קו התקשורת ושל ספק הקישור לאינטרנט. נושא שיש לתת עליו את הדעת, הוא רוחב הפס. איכות השירות שמקבלים גולשי האינטרנט, היא פונקציה של רוחב הפס בו מחובר האתר שלכם לאינטרנט. הדבר דומה לצינור מים. כשאתם שותים מים מהברז הביתי שלכם, הזרימה חזקה ואינכם מרגישים בבעיה, אך מה יקרה אם תפתחו בו-זמנית את כל הברזים אצלכם בבית? כל אחד יקבל זרימה חלשה יותר. זה בדיוק מה שקורה לאלפי גולשים, שמנסים להתחבר בו-זמנית לאתר בעל רוחב פס צר: זמני התגובה נעשים ארוכים. זה לא בגלל השרת, כי ה'צינור' גודלו קבוע. כשאתה משתמש בודד - כל הצינור שלך, כשיש שני משתמשים - לכל אחד יש חצי וכך הלאה. חוות שרתים רוכשות בדרך כלל חיבור מהיר ויקר יותר לשדרת האינטרנט (backbone), ובהם הבעיה פחותה. כדאי לעקוב באופן קבוע אחר ביצועי האתר ומספר המבקרים ביממה.

יתרון חשוב שיש להקמת שרת האינטרנט בתוך משרדי החברה, הוא גישה ישירה, שלא באמצעות האינטרנט, למסד הנתונים. כל המחלקות בחברה: הנהלה, שיווק, מחסן, משלוחים – כולן צריכות גישה מהירה ויעילה למסד הנתונים.

אחסנה חינם

ישנם אתרים רבים, המאפשרים אחסון אתר משלכם בחינם, או כמעט חינם. ברור שמדובר באתר לא מסחרי. אל תנסו להקים חנות פעילה באתר חינם (geocities ,xoom וכדומה). ברובם יהיה האתר שלכם מלווה בפרסומות המממנות את תחזוקת האתר. יש לשים לב בבחירת אתר האיחסון אם הוא תומך ב-ASP מי שאינו תומך ב-ASP - פסול. ככל שעולה הפופולריות של ASP, כך יש יותר אתרים (ואתרי חינם) שתומכים בו.

אתר מעניין הוא **http://www.brinkster.com** (לשעבר www.ewebcity.com). יש לו מספר יתרונות: הוא תומך ב-ASP, הוא חינם לחלוטין ואין בו פרסומות. מקום מצוין ללימוד ולהתנסות ב-ASP, מעבר לשימוש בשרת הביתי.

אתרים נוספים בהם ניתן לאחסן אתר ASP ללא תשלום:

http://webhosting.bootbox.net

http://www.websamba.com

http://www.domaindlx.com

http://www.webhostme.com

העתקת הפרויקט לשרת

במקרה בו שרת האינטרנט האמיתי אינו השרת מולו התבצע הפיתוח, נצטרך להעביר את הפרויקט כולו. למרבה המזל, Visual InterDev תומך בכך בצורה מלאה.

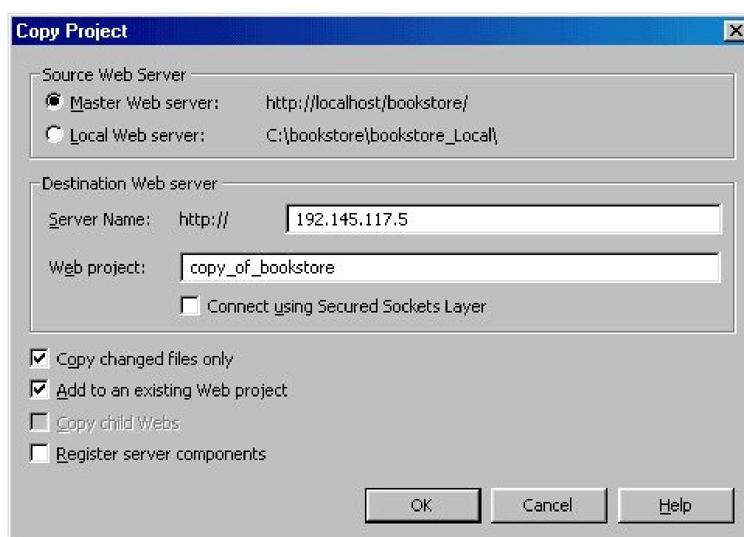
בחרו בתפריט הראשי: Project > Web Project > Copy Web Application, או פשוט לחצו על הסמל Copy Web Application בחלון Project Explorer, ותקבלו את חלון העתקת Copy Project.



שאלה ראשונה: מהיכן אתם רוצים להעתיק, מהעותק המקומי, או מעותק השרת הנוכחי? במצב Master בו אנו עובדים, אין הבדל, כי הרי כל שמירה מעדכנת את השרת. במצב Local, נבחר בדרך כלל להעתיק מהשרת, בו נמצאת הגירסה המעודכנת ביותר.

שאלה שנייה: להיכן להעתיק? כאן אתם נדרשים להקליד את כתובת ה-IP של שרת היעד, או את שמו, לדוגמה: www.sifroni.co.il. לאחר שבחרתם כתובת להעתקה, בחרו שם לפרויקט בשרת היעד.

שימו לב לאופציה המעניינת Copy changed files only. זו אפשרות שימושית ביותר לתחזוקה. כאשר האתר כבר הועתק ופועל, וברצוננו לעדכן את הקבצים שעברו שינוי מאז העתקתם לשרת, סמנו אפשרות זו. על פי השוואת תאריכים, יועתקו רק קבצים, שהשתנו מאז ההעתקה האחרונה.



חשוב לדעת, כי העתקת הפרויקט לשרת בשיטה זו תעבוד רק אם בשרת שלכם הותקן Front Page Extentions. אם לא – יהיה עליכם להעתיק ידנית את קבצי הפרויקט שלכם לשרת היעד, לתיקיה: c:\inetpub\wwwroot.

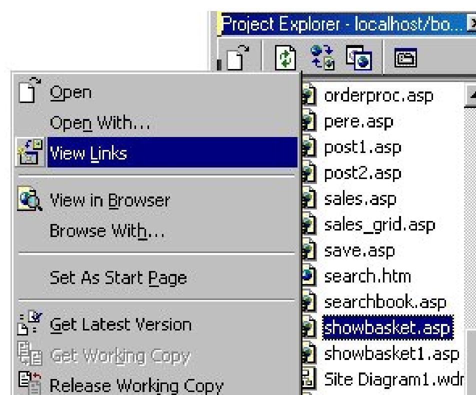
ומה עם ה- DSN?

כזכור לכם, DSN מכיל מידע הן על סוג מסד הנתונים והן על מיקומו הפיסי. בשרת, המיקום הפיסי כמעט בהכרח משתנה. לכן, כאשר אתם מעבירים את הפרויקט לשרת, יש ליצור בשרת DSN חדש. ומה עם דפי ה-ASP? האם יש צורך לשנות אותם? התשובה היא לא. בדיוק מאותה סיבה, שמקום הימצאו של מסד הנתונים טמון ב-DSN, אין צורך לשנות שום דבר בדפים שכתבנו. מובן שה-DSN החדש חייב להיות באותו שם, מפני שהוא מוזכר בכל אחד מהדפים שלנו.

תחזוקה שוטפת – שימוש ב- View Links

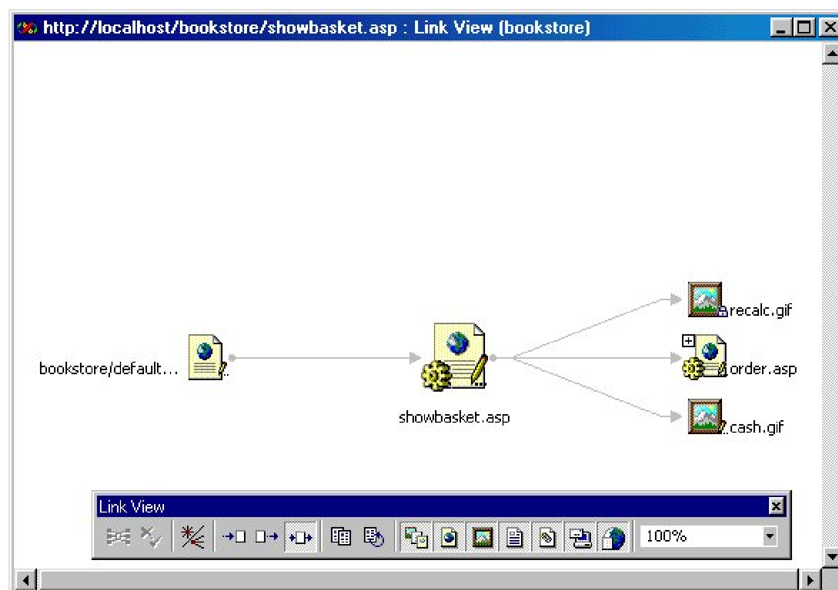
דיאגרמת הקישורים היא כלי מצוין לבדיקת תקינות של האתר. באופן בסיסי תפקידה פשוט ביותר: להראות דף (HTML או ASP) עם כל הקישורים שלו. אלה יכולים להיות קישורים לדפים אחרים, לאתרים מרוחקים, לקבצי מולטימדיה, ואפילו לתמונות ורקעים מהם מורכב הדף. עבור כל קישור ניתן לראות מה מצבו: תקין או לא.

הדרך לפתוח דיאגרמת קישורים: ב- Project Explorer לחצו לחיצה ימנית על שם הקובץ, ובחרו View Links.

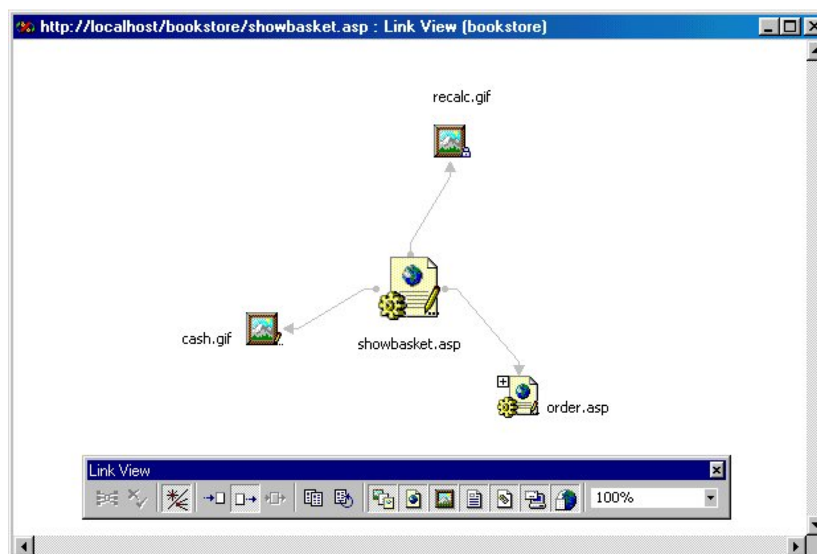


ייפתח חלון Link View ובו דיאגרמת הקישורים לדף זה. במרכז החלון אנו רואים את הדף המבוקש, ומסביבו את כל הקבצים הקשורים אליו. בדוגמה זו אנו רואים את הדף showbasket.asp המכיל שני לחצנים. לחצנים אלה מופיעים בדיאגרמה כקבצים קשורים. אם אחד הקבצים יהיה חסר, נראה אותו כסמל "שבור".

מדף זה יש קישור לדף ההזמנה order.asp. רואים יפה את הקישור. אם הדף לא היה קיים או לא היה זמין מסיבה כלשהי, היינו רואים סמל "שבור". שימו לב, לסימן + ליד הסמל של דף ההזמנה. ניתן ללחוץ עליו ולפתוח "עץ" נוסף של דף זה.



לעיתים נוח יותר לראות את הדיאגרמה בצורה מעגלית. כלומר: הדף הנבחר במרכז, ויתר הקבצים מסודרים סביבו במעגל. השתמש בסרגל הכלים Link View כדי לבחור את צורת ההצגה הברורה ביותר.



כדאי לבדוק מדי פעם את מפת הקישורים, ולוודא שכל ההפניות תקינות. נזכיר בהקשר זה גם את דוח הקישורים השבורים – Broken Link Report. תפקידו דומה, רק צורת ההגשה אחרת. בתפריט View בחרו Broken Link Report, ותקבלו בחלון Task List רשימה מפורטת של כל הקישורים הבעייתיים באתר שלכם. כדאי לבדוק אותם אחד לאחד.

פרק 21

מערכת הקבצים

תוכניתן ASP לא נזקק, בדרך כלל, למערכת הקבצים. כל הרעיון הוא, שכל מידע שיש לשמור – שומרים במסד הנתונים, ואין צורך להתעסק עם קבצי מערכת ההפעלה. אבל... לעיתים אין ברירה. יש להכין קבצי חיוב לחברות האשראי, יש להעביר חומר למערכות אחרות, יש לקלוט חומר ממערכות ישנות (legacy) ועוד. למרבה המזל, קיים בשרת אובייקט המיועד במיוחד לטיפול במערכת הקבצים - האובייקט FileSystemObject, שייך לתיקיה Scripting, המותקנת בשרת, והוא מטפל בקלות רבה ביצירת קבצים, כתיבה אליהם, קריאה מהם ועוד.

את כל הפעולות השימושיות נבצע בעזרת האובייקט File, אך כדי ליצור אותו, יש ליצור קודם את האובייקט FileSystemObject. כך:

set objFileSys=createObject("Scripting.FileSystemObject")

כשזה קיים, ניתן ליצור אובייקט File על ידי יצירת קובץ או פתיחת קובץ קיים.

התחביר ליצירת קובץ:

CreateTextFile(***Filename***, ***Overwrite***, ***Unicode***)

עליכם לספק את שם ומיקום הקובץ ליצירה, האם "לדרוך" עליו אם כבר קיים (ברירת המחדל: לא), והאם להשתמש בשיטת unicode לכתיבת הקובץ (ברירת המחדל: לא).

דוגמה:

set myFile=objFileSys.CreateTextFile("c:\users.log")

כאמור, ניתן כמובן לפתוח גם קובץ קיים. להלן התחביר לפתיחת קובץ:

OpenTextFile(*Filename IOMode*, *Create*, *Format*)

Filename – שם ומיקום הקובץ.

IOMode – צורת הפתיחה: 1– לקריאה, 2– לכתיבה, 8– להוספה.

Create – אם אין קובץ כזה, האם ליצור חדש?

Format – שיטת הפתיחה: 0 (ברירת מחדל)– ASCII. ‎–1 unicode.

דוגמה:

set myFile=objFileSys.OpenTextFile("c:\users.log",1,0)

כעת, כשיש לנו אובייקט File, נכתוב לתוכו טקסטים. תחביר הכתיבה:

FileObject.Write *Text*

דוגמה:

myFile.Write "Today is: " & date

ניתן להשתמש גם ב- WriteLine, שכותבת את הטקסט, ומוסיפה מעבר לשורה חדשה בסופו.

כשיודעים לכתוב, רצוי גם לדעת לקרוא. כך קוראים טקסט מתוך קובץ:

MyText=myFile.ReadLine

הטקסט בשורה עליה עמד הסמן בקובץ, עובר לתוך המשתנה MyText.

דוגמה לשימוש במערכת הקבצים

נבנה מערכת, הכוללת שלושה דפים. הדף הראשון יהיה דף כניסה (login) למערכת. בדף זה יקליד המשתמש שם וסיסמה. אם השם או הסיסמה שגויים – יש לרשום אותם בקובץ, עם תאריך ושעה, ולא לאפשר כניסה. אם הם נכונים, יש לכתוב בקובץ רק "כניסה נכונה" ותאריך, ולהעביר את המשתמש למסך היישום. במסך זה נבצע קריאה מהקובץ ונראה את כל השורות הכתובות בו.

דף ראשון - file_login.asp

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<FORM action="file_check.asp" method=post id=form1 name=form1>

<B>Please enter your name and password.<br>
Hint: the name is 'dani', and the password is '1234'.<br><br>
Name: <INPUT id=text1 name=name ><BR>
Password:<INPUT id=text2 name=pass ><BR>
</B><BR>
```

```
<INPUT type="reset" value="Reset" id=reset1 name=reset1 style="HEIGHT:
24px; WIDTH: 98px">      
<INPUT type="submit" value="Submit" id=submit1 name=submit1
style="HEIGHT: 24px; WIDTH: 106px"><BR>
</FORM>

</BODY>
</HTML>
```

כך הוא ייראה בדפדפן:



אני מניח שדף זה אינו צריך הסבר. קולטים שם וסיסמה, ומעבירים אותם לדף הבא.

דף שני - file_check.asp

```
<%@ Language=VBScript %>
<%
dim name, pass
name=Request.Form("name")
pass=Request.Form("pass")

set objFileSys=createObject("Scripting.FileSystemObject")
if not objFileSys.FileExists("c:\users.log") then
 set myFile=objFileSys.CreateTextFile("c:\users.log")
else
 set myFile=objFileSys.OpenTextFile("c:\users.log",8)
end if

if name="dani" and pass="1234" then

myFile.WriteLine "Valid entry. Date: " & date & " Time: " & time
myFile.Close
Response.Redirect "file_welcome.asp"

else
myFile.Write "Invalid entry. Date: " & date & " Time: " & time
myFile.WriteLine " Name: " & name & " Password: " & pass
myFile.Close

Response.Write "Invalid name or password. <br>"
Response.Write "Press 'back' and think again."
end if

%>
```

הסבר:

1. קולטים את הפרמטרים: name, pass.
2. בודקים אם קיים קובץ הנתונים. אם קיים – פותחים אותו לכתיבה נוספת. אם לא קיים – יוצרים אותו.
3. אם השם והסיסמה נכונים – כותבים שורה חדשה ובה המילים: Valid Entry והתאריך. מפנים את הגולש לדף file_welcome.asp.
4. אם השם והסיסמה לא נכונים – כותבים שורה ובה השם והסיסמה שנמסרו, ותאריך ההקלדה. כותבים הודעה למסך: הנתונים אינם נכונים, אנא חזור ותקן.

דף שלישי - file_welcome.asp

לדף זה מגיע הגולש רק אם הקליד שם וסיסמה נכונים. כאן נקרא את הנתונים מתוך הקובץ, ונציג אותם על המסך.

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY bgColor=deepskyblue>
<P align=center><FONT face="Comic Sans MS" size=6
color=yellow><STRONG>Welcome to my
application! </STRONG></FONT></P>

<%
set objFileSys=createObject("Scripting.FileSystemObject")
if objFileSys.FileExists("c:\users.log") then
 set myFile=objFileSys.OpenTextFile("c:\users.log",1,0)
 while not myFile.AtEndOfStream
 mytext=myFile.ReadLine
 Response.Write mytext & "<br>"

 wend

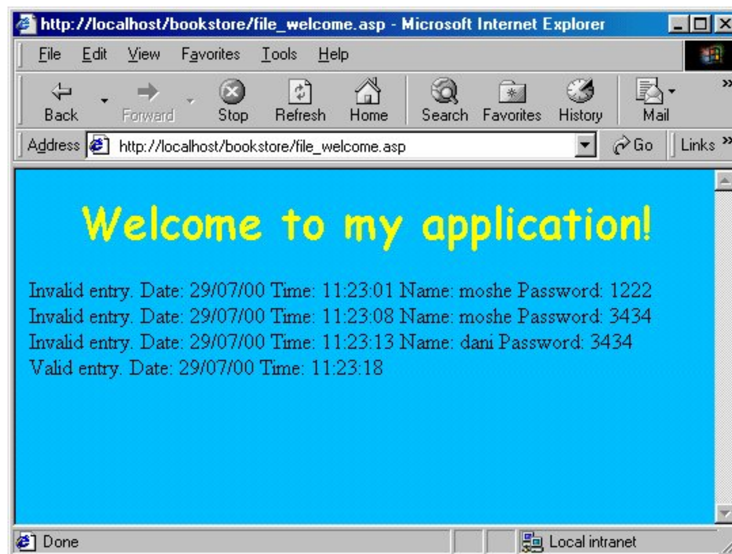
else
 Response.Write "The file c:\users.log not found. Maybe no user entered?"
end if
%>

</BODY>
</HTML>
```

הסבר:

1. בדוק אם הקובץ קיים. אם לא קיים – תן הודעה מתאימה.
2. אם קיים, פתח אותו לקריאה.
3. עבור על כל הקובץ, וכל עוד לא הגעת לסופו...
4. קרא שורה וכתוב אותה כפי שהיא למסך.

בזמן ריצה, החלון ייראה בערך כך :



שימושים נוספים לאובייקט File

האובייקט File, משמש גם לטיפול בקבצים ברמת מערכת ההפעלה. כלומר : לא קריאה וכתיבה של נתונים כפי שעשינו עד כה, אלא מידע על גודל הקובץ, מיקומו בדיסק וכדומה. ופעולות מערכת הפעלה כמו העתקה, מחיקה וכדומה.

כדי לקבל גישה אל האובייקט File, נשתמש בפונקציה GetFile, כך :

```
set fs=Server.CreateObject("Scripting.FileSystemObject")
myfile=fs.GetFile("c:\users.log")
```

ואז נוכל לקרוא את המאפיינים (properties) ולהשתמש בשיטות (methods) של האובייקט.

רשימת המאפיינים של האובייקט File :

Property - מאפיין	הסבר
Attributes	תכונות הקובץ. יכול להיות חיבור של התכונות הבאות: 0 – רגיל 1 – קריאה בלבד 2 – חבוי 4 – קובץ מערכת
DateCreated	תאריך יצירת הקובץ
DateLastAccessed	תאריך גישה אחרונה לקובץ
DateLastModified	תאריך שינוי אחרון של הקובץ
Dtive	שם הכונן בו נמצא הקובץ
Name	שם מלא של הקובץ
ParentFolder	התיקיה בה נמצא הקובץ
Path	המסלול לקובץ
ShortName	שם קצר, בפורמט 8.3
ShortPath	מסלול קצר, בפורמט 8.3 (שמות התיקיות יקוצצו לאורך 8)
Size	גודל הקובץ
Type	סוג הקובץ

רשימת השיטות של האובייקט File :

Method - שיטה	הסבר
Copy	העתק את הקובץ ממקום אחד לשני
Delete	מחק את הקובץ
Move	העבר את הקובץ ממקום אחד לשני
OpenAsTextStream	פתח את הקובץ והחזר אובייקט TextStream, אליו ניתן לכתוב ולקרוא.

דוגמה :

```
set objFileSys=createObject("Scripting.FileSystemObject")
set myfile=objFileSys.GetFile("c:\users.log")
Response.Write "<br>"
Response.Write "File Name: " & myfile.name & "<br>"
Response.Write "File size: " & myfile.Size & "<br>"
Response.Write "File create date: " & myfile.datecreated & "<br>"
```

פרק 22

לחיות בסטייל – עיצוב אתרים מקצועיים בעזרת סגנונות ותבניות

"עקביות היא סימן ההיכר של המקצוען", אמר פעם מישהו, וידע על מה הוא מדבר. קל מאוד להבחין בין אתר שבנה בעל מקצוע, לבין אתרי חובבים. אצל המקצוענים תהיה, בדרך כלל, אחידות מראֶה בין הדפים. הרקע יהיה זהה בכל הדפים, הכותרת תהיה באותו גודל, גופן וצבע בכל הדפים. הסמלים (Icons), יהיו בעלי עיצוב דומה וקו משותף. ואילו אצל החובבים – בדרך כלל יהיה בלגן, כל דף בצבע אחר, כותרות בכל צבעי הקשת, ושפע של קישורים, שאינם מובילים לשום מקום.

בפרק זה נעסוק בדפי סגנון – Cascading Style Sheets, ובקיצור CSS. זהו אמצעי טכני פשוט, שנועד לעזור לנו להשיג את המטרה – אתר בעל מראה עקבי. הכוונה אינה דווקא לאתר משעמם וסולידי, המתאים לבנק או לסוכנות ביטוח. עקביות תהיה תכונה מבורכת גם באתר קופצני ועליז.

תפקיד חשוב לא פחות של דפי הסגנון, הוא בקרה משופרת על עיצוב הדף. HTML, אם לא שמתם לב, הוא כלי עיצוב מוגבל ביותר. דפי הסגנון משלימים אותו, ונותנים בידי המעצב שליטה ברמת פירוט גבוהה הרבה יותר.

מהם דפי סגנון?

דף סגנון, בהגדרתו, הוא קובץ המכיל הגדרות של תגי HTML. לדוגמה, ניתן להגדיר מחדש את התג <H1>. אנו רוצים שהמשמעות של תג זה תהיה: גודל אותיות 20 נקודות, גופן "David", צבע אדום וקו תחתון. זוהי הגדרה הרבה יותר מפורטת מזו שמאפשר HTML רגיל.

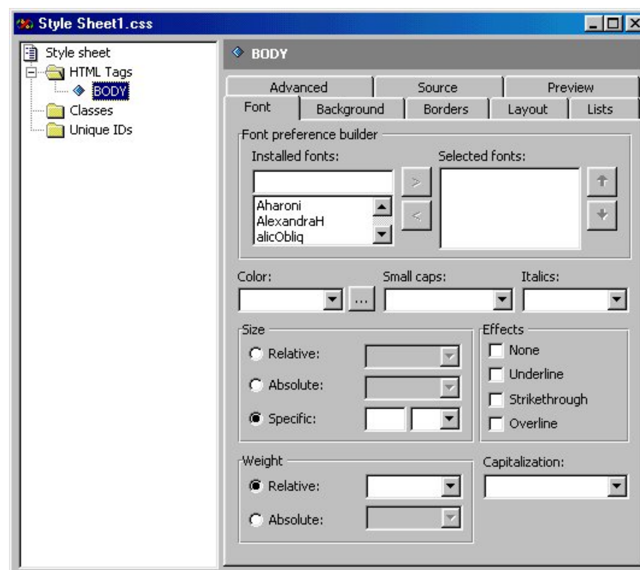
הנה קטע מתוך CSS, המבצע הגדרה זו:

```
<STYLE>
H1
{
   COLOR: red;
   FONT-FAMILY: David;
   FONT-SIZE: 20pt;
   TEXT-DECORATION: underline
}
...
</STYLE>
```

מרגע שהגדרנו סגנון זה, בכל מקום בו נשתמש בתג <H1>, תהיה השפעתו זהה להגדרה החדשה. נגדיר מספר סוגי כתב, ונקפיד שכל טקסט באתר שלנו יהיה שייך לאחת ההגדרות. לדוגמה: טקסט רגיל, כותרת משנה, כותרת ראשית. כך יצרנו מקום מרכזי, בו ניתן לשלוט על חזותו של כל האתר! רוצים להחליף את צבע הכותרות? שנו אותו בדף הסגנון, הוא יתחלף מייד בכל הדפים! רוצים לשנות את צבע או תמונת הרקע של כל 50 הדפים באתר? החליפו את הרקע בדף הסגנון, וראו כיצד כל האתר שלכם מחליף רקע! (מידע נוסף בספר "**HTML 4 למפתחי אתרים באינטרנט**" בהוצאת **הוד-עמי**).

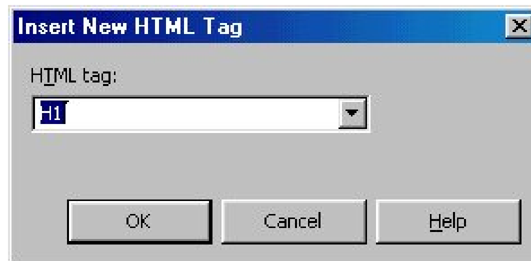
יצירת דף סגנון

לחצו לחיצה ימנית על שם הפרויקט, ובחרו Add > Cascading Style Sheet. תקבלו דף דומה לזה:



בחלק השמאלי, ניתן לראות את המבנה ההיררכי של הדף. בחלק הימני, מופיע דף המאפיינים של האלמנט הנבחר. בדוגמה שלנו, הסמן הוצב על התג <BODY> (בצד שמאל), לכן דף המאפיינים מראה את כל המאפיינים החדשים של תג זה. לחצו לחיצה ימנית על התיקיה HTML Tags, ובחרו באפשרות: ...Insert HTML Tag.

תקבלו חלון Insert New HTML Tag, פתחו את תיבת הגלילה ובחרו בתג H1.



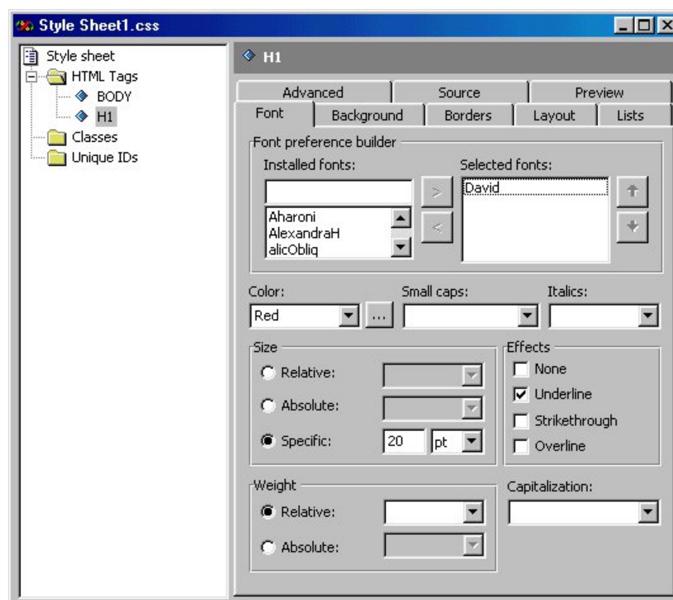
כעת נוסף H1 לרשימת תגי HTML, ואתם מוזמנים לשנות את מאפייניו בחלון המאפיינים, בצד ימין.

חלון המאפיינים בנוי ממספר כרטיסיות. בואו נטייל על פניהם, ונסביר את תפקיד כל אחת מהן.

Font

כאן נקבעים מאפייני הגופן. בוחרים גופן, צבע, גודל, אפקטים (מודגש, נטוי, קו תחתון וכדומה) ועוד.

בחרו עבור הדוגמה שלנו באפשרויות שהוזכרו למעלה: גודל אותיות 20 נקודות, גופן "David", צבע אדום וקו תחתון. אך מה קורה, אם אתם רוצים להציג את הכותרות שלכם בגופן "David", אך במחשב הלקוח, לא מותקן גופן זה? יש אפשרות לתת רשימת גופנים. דף הסגנון ינסה לטעון תחילה את הגופן הראשון ברשימה, אם לא מצא – יעבור הלאה לגופן הבא בתור וכך הלאה. השתמשו בחיצים בצד ימין כדי לקבוע את סדר העדיפות של הגופנים.



Background

כאן תוכלו לקבוע את צבע הרקע, תמונת הרקע ואת המיקום, בו תוצב תמונת הרקע על החלון.



הערה:

ניתן להגדיר לכל תג תמונת רקע משלו, אבל זוהי לא הכוונה, בדרך כלל. לרוב, המקום של תמונת הרקע הוא בתג BODY.

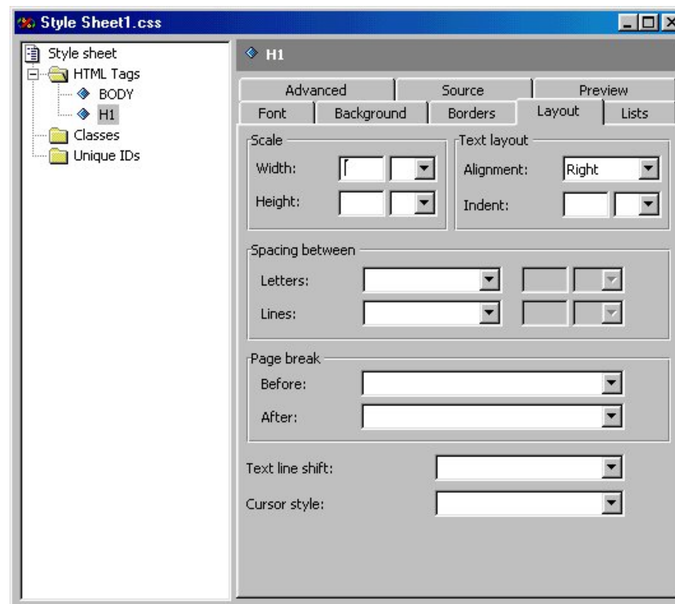
Borders

קביעות הנוגעות לגבולות הטקסט : סוג הגבול (יחיד, כפול, וכדומה), צבע הגבול, גודל שוליים (מרחק הגבול מהטקסט) וכן הלאה.

Layout

קביעות הנוגעות לעריכת הטקסט ומיקומו על פני הדף.

בוני אתרים בעברית: שימו לב לאפשרות החשובה Alignment. אם תבחרו Right, כמו במסך, ייושר הטקסט שסימנתם באופן אוטומטי לימין. זה חוסך הרבה מאוד עבודה.



שימו לב גם לאפשרות Cursor Style (למטה). כאן ניתן לקבוע את צורת הסמן, כאשר עומדים על הטקסט המסומן. מעניין, אך עלול לבלבל!

Lists

מטפל בעיצוב רשימות. מאפשר לקבוע, בין היתר, את תמונת ה- Bullet. זוהי אותה תמונה קטנטונת, המקדימה את סעיפי הרשימה. זוהי עוד דוגמה מצוינת לטענתי בדבר העקביות: אתר שבו כל הרשימות מופיעות עם אותו סמל, מעניק לגולש אווירת ביטחון, אמינות וסדר. גם כאן נרגיש ביתרון האדיר של CSS: החלפת Bullet בכל האתר, היא עניין של שניות, כי מחליפים רק במקום אחד.

Advanced

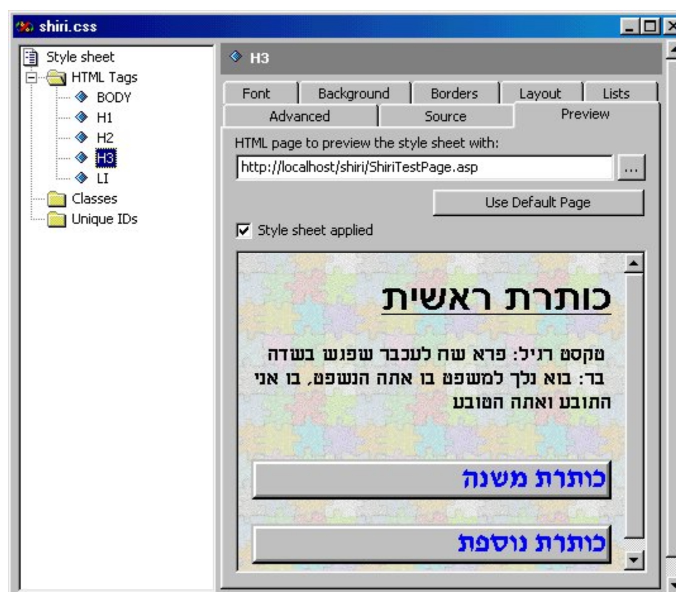
קביעות מתקדמות. כברירת מחדל, מוצגים האובייקטים בדף לפי סדר הופעתם בדף HTML. ניתן לעקוף קביעה זו, על ידי מתן קואורדינטות מדויקות לכל אובייקט. שימו לב לאפשרות המעניינת Z Order. הכוונה היא לסדר האובייקטים בציר המאונך למסך. כלומר: מי מסתיר את מי? אובייקט בעל Z Order נמוך יוסתר על ידי אובייקט בעל Z Order גבוה. באנלוגיה זו ישנם שלושה צירים: X, Y המוכרים, הם אורך ורוחב, ואילו ציר Z יוצא מהמסך לכיוון המתבונן...

Source

כאן ניתן לראות את הקוד, שהתקבל מהקביעות בכרטיסיות השונות. חשוב לזכור, שבעצם מדובר בקובץ טקסט רגיל. כל החלונות היפים אינם אלא ממשק משתמש נוח. התוצאה הסופית היא מה שכתוב בחלון זה. **אי אפשר לשנות את הקוד בחלון זה!** שימו לב לכך, שכאשר אתם מסמנים תג מסוים בעץ המבנה, החלק המתאים לו בחלון הקוד, נכתב באות מודגשת.

Preview

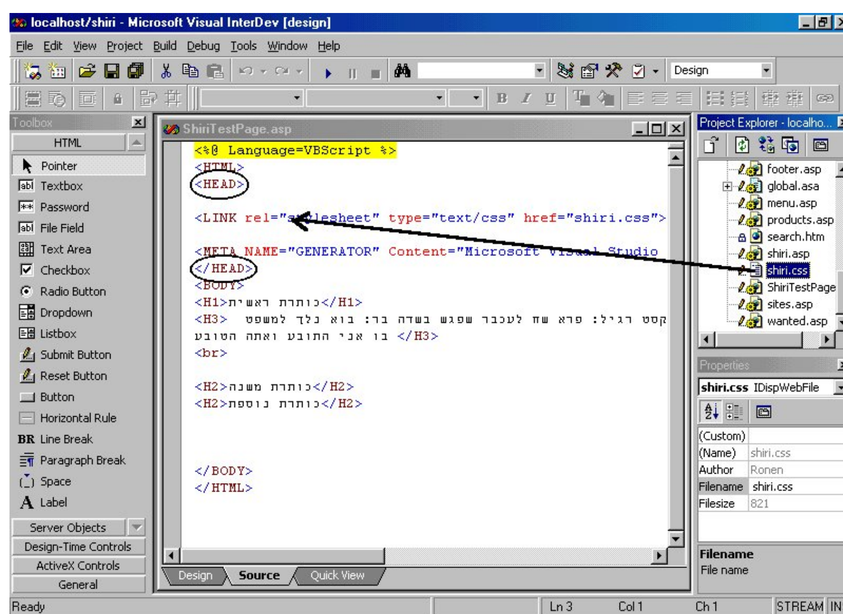
כרטיסיה חשובה ביותר, המראה לכם בזמן אמת, כיצד נראית תוצאת עבודתכם. שחקו עם תיבת הסימון Style Sheet Applied, כדי לראות כיצד נראה התג לפני השינוי שלכם, ואחריו. הדף המוצג הוא דף סטנדרטי המכיל תגי HTML רבים, ובדרך כלל תקבלו אינדיקציה טובה למדי לגבי התוצאה באתר הפרטי שלכם. אבל יש כאן אפשרות מדהימה נוספת: לקבוע דף צפייה מקדימה משלכם! לחצו על לחצן שלוש הנקודות, ובחרו דף כלשהו שלכם. דף זה ישמש לבדיקה מהירה של שינויי הסגנון שביצעתם. זה יכול להיות דף אמיתי מתוך הפרויקט, או דף ניסיון, המיועד לכך.



שימוש בדפי סגנון

לאחר שיצרנו את דף (או דפי) הסגנון, אנו רוצים להשתמש בהם. כלומר, להחיל אותם על דפים קיימים. הנה הדרך הקלה ביותר:

1. פתחו את הדף עליו אתם רוצים להחיל סגנון.
2. פתחו מעט מקום בין התגים <HEAD> ו- </HEAD>.
3. גררו את דף הסגנון מה- Project Explorer ישירות לתוך הדף שלכם, בדיוק בין התגים הנזכרים למעלה. כך:



שימו לב, חובה לגרור את הדף אל בין התגים <HEAD> ו- </HEAD>. אחרת – לא עובד! היכן בין שני התגים? זה כבר לא משנה.

תופיע שורה דומה לזו:

<LINK rel="stylesheet" type="text/css" href="Style Sheet1.css">

ניתן כמובן גם לכתוב אותה ידנית, ולערוך אותה במקרה של שינויים.

הריצו את הדף שלכם, וראו את התוצאות.

כתיבת סגנונות בתוך הדף

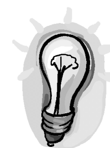
עד עכשיו ראינו כיצד לכתוב דפי סגנון עצמאיים. כלומר: דף שלם וסגור שתפקידו להיות דף סגנון, ודפים אחרים, רגילים, המכילים הפניה לאותו דף סגנון. כך מקובל ורצוי לעבוד. אבל זו לא הדרך היחידה. ניתן לכתוב פקודות סגנון גם בתוך דף רגיל. לדוגמה, ניתן לכתוב כך:

```
<STYLE>
H1
{ FONT-SIZE: 24pt;  }

</STYLE>
```

אין בכך שום ייחוד, כבר כתבנו דברים כאלה. השאלה המעניינת היא: מה קורה להגדרה כזו בדף המקושר לדף סגנון, כמו שבנינו בעמודים הקודמים, המכיל כבר הגדרת סגנון עבור <H1>?

התשובה היא, שההגדרה המקומית מתגברת (Override) על ההגדרה הכללית, אך בכל מקום בו לא נגעה ההגדרה המקומית – נשארת ההגדרה הכללית. לדוגמה: ההגדרה הכללית עבור <H1> היתה גודל אותיות 20 נקודות, גופן "David", צבע אדום וקו תחתון. ההגדרה המקומית אומרת רק גודל 24 נקודות. ומה עם השאר? יישאר לפי ההגדרה הכללית. זו הסיבה לשם Cascading – ההגדרה מתגלגלת מדף לדף.



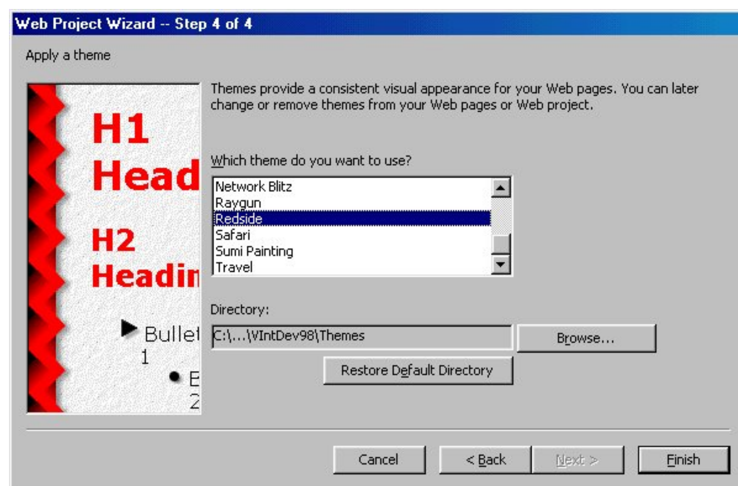
טיפ:

עורך דפי הסגנון עובד רק בדפי סגנון עצמאיים, ואינו יודע לטפל בסגנונות המוטמעים בתוך דפים רגילים. טריק פשוט לעקוף מגבלה זו, הוא ליצור דף סגנון עצמאי, וליצור בו את הסגנונות שרוצים. בחלון Source מעתיקים את הקוד (Copy), עוברים לדף בו רוצים לכתוב את הסגנון, ומדביקים (Paste) את הקוד. פשוט, לא?

עיצוב נושאי - תימה (Theme)

תימה, לפי הגדרתה במילון, הוא שיר נושא, מוטיב החוזר על עצמו. בהשאלה לענייננו, Theme הוא אוסף מתואם של דפי סגנון, המהווים מערכת של רקעים, תמונות, גופנים ושאר אלמנטים ויזואליים, המרכיבים ביחד את עיצובו הכללי של האתר, ושהוגדרו מראש.

נפגשנו בעבר עם Themes בשלב יצירת הפרויקט, אלא שאז דילגנו עליהם בדרך לדברים חשובים יותר. כעת ניזכר בשלב 4 של יצירת הפרויקט בו ניתנה לנו האפשרות לבחור Theme מתוך אפשרויות רבות.

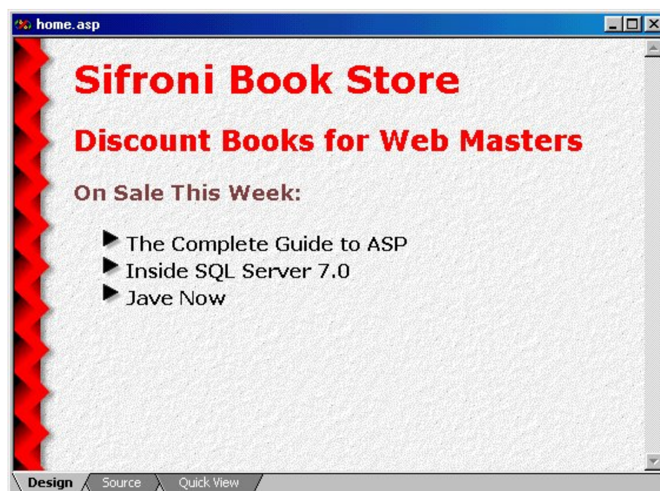


בדוגמה זו בחרתי ב-Theme בשם Redside. טיילו על פני רשימת ה-Themes והתבוננו בחלון הדוגמה. לאחר אישור הדוגמה ויצירת הפרויקט, שימו לב לחלון Project Explorer: נוצרה תיקיה מיוחדת בפרויקט עבור ה-Theme, ובה קבצים רבים המרכיבים אותו. כאשר ניצור דף כלשהו, נקבל את הדף כשהוא כבר מכיל אלמנטים רבים של עיצוב. בין התגים <HEAD> ל- </HEAD> נראה לא פחות מ-4 שורות של קישור לדפי סגנון!

נעבור למצב Design, ונוסיף כותרת <H1>, כותרת משנה <H2>, ורשימה (Bulleted List).

כותבים את השורה ישירות לחלון Design, מסמנים את הטקסט, ובוחרים, בתיבת הסגנונות הנגללת הנמצאת בסרגל הכלים, את הסגנון הרצוי.

התוצאה נראית כך:

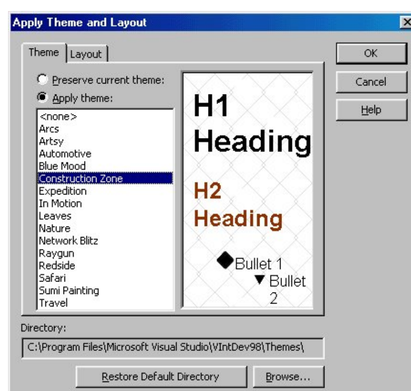


כל מתכנת HTML יסכים שזו תוצאה מדהימה, ביחס לכמות הזמן שהושקעה בעיצוב (אפס!).

הוסיפו לכך את העובדה, שכל דפי האתר עומדים להיראות כך, וקיבלתם עסקת חבילה משתלמת ביותר: עיצוב נקי, מסודר ועקבי לאורך כל האתר. כדאי.

החלת Theme (סגנון) על פרויקט קיים

ראינו שקביעת ה-Theme התבצעה בעת יצירת הפרויקט. אבל מה לעשות אם נזכרתי מאוחר והפרויקט כבר קיים? ומה לעשות במקרה שהלקוח לא משתגע על ה-Theme שבחרתי, ורוצה משהו אחר? ניתן להחיל Theme גם על פרויקט קיים. לחצו לחיצה ימנית על שם הפרויקט, ובחרו Apply Theme and Layout. יתקבל חלון Apply Theme and Layout:



בחרו ב-Theme הרצוי ואשרו. ה-Theme יופעל על כל דפי הפרויקט. המשימה שתישאר לכם היא רק לעבור על הדפים, ולקבוע עבור כל פיסקה או משפט – לאיזה סגנון הם שייכים.

תאימות בין דפדפנים

אי אפשר לסכם פרק העוסק בדפי סגנון, ללא מילה של אזהרה. CSS הוא נושא חדש יחסית, ולא כל הדפדפנים תומכים בו. גרוע מכך, דפדפנים שונים מטפלים ב-CSS באופן שונה!

בעיה ראשונה, שיש להתייחס אליה, היא קהל המשתמש בדפדפנים ישנים, שאינם תומכים ב-CSS כלל. CSS 1.0 נתמך על ידי Microsoft Explorer מגירסה 3.0 ומעלה, ועל ידי Netscape Communicator מגירסה 4.0 ומעלה. גרסאות קודמות לכך אינן תומכות ב-CSS, והדפים שהכנתם עלולים להיראות בהם רע מאוד. הדילמה היא: האם לכתוב אתר מרשים, מסוגנן ועקבי, עבור רוב הגולשים, או אתר פשוט יותר לכולם.

גישה אחת להתמודדות עם הבעיה, היא להגביל את השימוש ב-CSS לתגי HTML סטנדרטיים בלבד, כמו: B, Li, ... H2, H3 וכדומה. גם אם לא יעבוד הסגנון שבנינו, עדיין יוצג משהו קריא. לא בגופן שבחרנו, לא בצבע שרצינו, אבל האתר ימשיך לעבוד. כך, גולשים בעלי דפדפנים מודרניים, ייהנו מחווית גלישה עשירה ומהנה, ואלה שלא – לא יינזקו.

גישה אחרת, היא להחזיק שתי גרסאות של כל דף: האחת גרסת CSS, והשנייה גירסה פשוטה יותר. עליכם לבדוק איזה דפדפן יש לגולש (כיצד?), ולשלוח לו את הדף המתאים. מובן שיש כאן בעיית תחזוקה קשה: כל שינוי בדף צריך לבצע פעמיים.

בעיה מסוג אחר, היא דפדפנים שכן תומכים ב-CSS, אבל קצת אחרת. ישנם הבדלים קלים לא רק בין המתחרים הקבועים Netscape ו-Internet Explorer, אלא גם בין גרסאות שונות של אותו מוצר! עיקר ההבדלים הם בנושאים של מיקום מדויק, מרווחים בין שורות, גבולות ושוליים. אין ברירה, יש לבדוק את האתר בדפדפנים אחדים, חדשים וישנים, ולוודא שהעיצוב המדהים שלכם אכן יגיע ליעדו בשלום.

מומלץ להציב הערה בדף הפתיחה של האתר ולציין בה באיזה דפדפן ובאיזו גירסה מומלץ להשתמש. לדוגמה: "מומלץ לצפייה בדפדפן Internet Explorer מגירסה 5 ומעלה".

פרק 23

משימת אבטחה

מהן הסכנות?

מכל המשימות המוטלות עליכם במסגרת בניית יישום אינטרנט, אבטחת האתר היא המורכבת והקשה ביותר לביצוע. טעות נפוצה היא לדחות את הטיפול בבעיה עד לסיום העבודה על האתר, "קודם שיעבוד, אחר כך נדאג לאבטחה". ישנם גורמי אבטחה רבים ופשוטים, שכדאי שיעבדו כבר מהיום הראשון, ואין הכוונה כאן ל-Fire Walls (קירות אש) או למפתחות אסימטריים. בהשאלה ניתן לומר כך: לפני שאתם שוכרים את סיירת מטכ"ל לשמור על ביתכם, מוטב שפשוט תסגרו את הדלת...

הסכנות האורבות לאתרכם הצעיר הן רבות. גם בכך חלק מהקושי, בעיקר אם מדובר באתר מסחרי, בו יש תנועת כספים וסחורות. כדאי לערוך רשימה של הסכנות, כדי שנדע מאיזה כיוון להתגונן. ניתן לסווג את הסכנות לאתר ל- 4 גורמים והם:

1. **מרגלים** - הכוונה לא דווקא לפגיעה פיסית באתר, אלא רק לקבלת מידע הנמצא בו: כמות מכירות, אילו מוצרים נמכרים, פרטים על לקוחות ועוד. חשוב מאוד למנוע ממרגלים כאלה גישה לאתר.
2. **מזיקים** - אלה יכולים להיות מתחרים עסקיים, שליחיהם, או סתם כאלה שנהנים לגרום נזק. אלה ישמחו למחוק לכם את טבלת המוצרים, או לכתוב את שמם על דף הבית שלכם.
3. **גנבים** - מטרתם של אלה פשוטה: להרוויח כמה לירות ללא מאמץ. בקטגוריה זו נמצאים: מעתיקי כרטיסי האשראי, מזמיני סחורה בכרטיס לא שלהם וכדומה.
4. **סכנות מהבית** - כפי שיודע כל קצין בטחון, עובדי החברה הם סיכון ממשי: יש להם גישה פיסית (למשל, מפתחות המשרד), הם מבינים בנושא האתר, יש להם גישה למחשב, והם רק בני אדם. אין צורך לפתח פרנויה, אבל יש צורך לפתח מערכת הרשאות מסודרת בחברה.

מהן הדרישות ממערכת האבטחה?

1. **חסימת כניסה ללא מורשים** - אנשים שאינם מורשים, מתוך החברה או מחוצה לה, לא יוכלו לבצע פעולות מסוימות.
2. **אבטחת מידע** - המידע הרגיש שמאוחסן בשרת לא ידלוף החוצה.
3. **אבטחת התקשורת** - הגנה על המידע המועבר בתקשורת בין שרתים מפני קריאה של אנשים לא מורשים.
4. **אבטחת הגולשים** - אבטחת פרטיותם של לקוחותינו. מספר כרטיס האשראי הוא הדוגמה הבולטת ביותר, אבל גם על פרטים אחרים, פחות חשובים, עלינו להגן.
5. **זיהוי הגולשים בוודאות** - מניעת "מתחזים" מלקבל שירותים מהאתר.
6. **שרידות** - חסינות מפני התקפות וניסיונות נזק מבחוץ. במקרים רבים, נפילת שרת האינטרנט פירושה השבתת החברה כולה, נזק כספי גדול ופגיעה במוניטין. לכן, ננסה בכל האמצעים העומדים לרשותנו למנוע נפילה כזו. מצד שני נדאג גם למנגנוני התאוששות וגיבוי למקרים של השבתה כוללת.

הגנה על השרת – אבטחה ב- NT

שלב ראשון בבניית מערך ההגנה שלנו, צריך להיות הגנה ברמת מערכת ההפעלה. כאשר השרת שלנו מחובר לרשת האינטרנט, הוא חלק מרשת עולמית מבוססת TCP/IP. תהיה זו הנחה אופטימית מדי להניח, שכל מי שירצה להתחבר למערכת שלנו, יעשה זאת דרך הממשקים שאנחנו בנינו. ישנן המון דרכים אחרות, וחובתנו לחסום אותן. NT מספקת רמה גבוהה מאוד של אבטחה, אך גם מספר מדהים של פרצות. אם תשתמשו בה בתבונה תהיו מוגנים ברמה טובה מאוד. אלו התכונות העיקריות ההופכות את Windows NT לבטוחה:

1. זיהוי משתמשים.
2. קבוצות משתמשים.
3. מעקב אחר פעילות (Auditing).
4. מערכת קבצים מאובטחת – NTFS.

בפרק זה תמצאו טיפים רבים בנושא אבטחת NT בכלל, ואבטחת IIS בפרט. רוב הטיפים מיועדים למשתמשי NT מנוסים. אם אינכם כאלה, עבדו בזהירות או היעזרו במומחים. נושא האבטחה הוא רחב ומורכב ודי בו כדי למלא ספרים. גם אם אינכם System Administrators מדופלמים, תקבלו כיוון ומושג כללי ממה יש להיזהר.



טיפ:

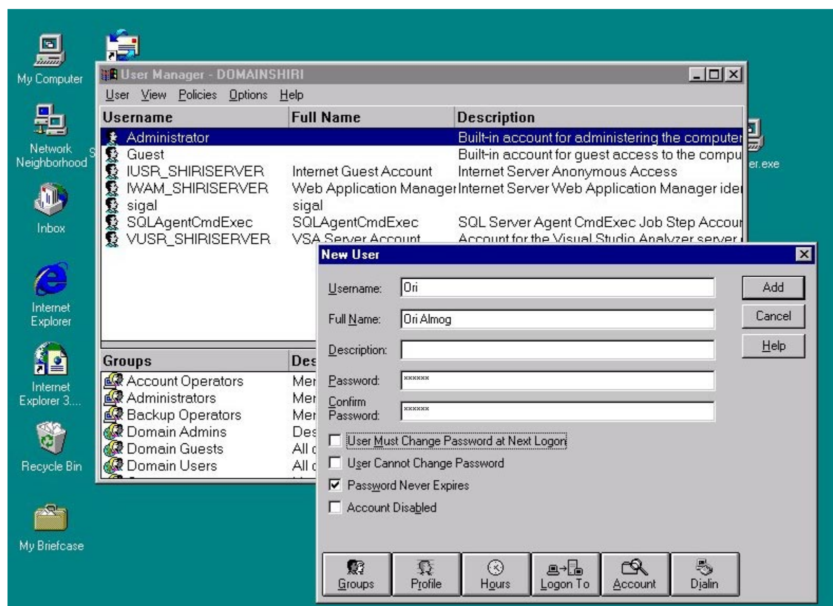
לפני שניגשים להפעיל את מנגנוני ההגנה של NT, ודאו שבשרת שלכם מותקנת הגירסה המעודכנת ביותר, כולל Service Pack האחרון ו- Hot Fixes האחרונים. אלה כוללים במקרים רבים תיקון פרצות אבטחה שהתגלו. תוכלו להוריד אותם מהכתובת:

http://www.microsoft.com/security

חשבונות משתמשים – User Accounts

כל משתמש מחשב ברשת, מכיר את חלון הכניסה לרשת. עליו להקליד שם וסיסמה, ורק אם אלה נכונים – הוא מצליח להתחבר. עבור כל משתמש מחזיק השרת "חשבון", מעין כרטיס ובו פרטי המשתמש: שם, סיסמה, שם מלא ועוד פרטים, כפי שניתן לראות בחלון New User.

מנהל הרשת יכול להוסיף משתמשים, למחוק אותם, ולקבוע את תכונותיהם בעזרת מנהל המשתמשים – User Manager for Domains.



תרשים 22.1: תמונת User Manager

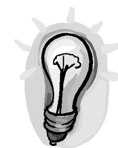
חשוב לדעת, כי שמות המשתמשים מיועדים לנוחיות השימוש בלבד. לכל משתמש יש גם קוד סודי - **Security ID** או בקיצור **SID**. מספר זה אינו נראה לעין, גם לא לעיני מנהל המערכת. מספר חדש מוקצה לכל משתמש חדש. לכן, אם נמחק משתמש וניצור

אותו מחדש, גם אם באותו שם ואותה סיסמה, זה יהיה משתמש חדש. לא ניתן יהיה לשחזר את המשתמש שנמחק.

כאשר תקימו שרת אינטרנט בעזרת IIS, כדאי להתקין את השרת בתצורת Stand Alone. כלומר, זה לא יהיה השרת אליו ניגשים המשתמשים לאישור סיסמאותיהם. הדבר נעשה ראשית כדי לשפר את רמת האבטחה, ושנית כדי להוריד את העומס על השרת.

את רוב המשתמשים יוצר מנהל המערכת. אך יש גם יוצאים מכלל זה. מספר יישומים יוצרים לעצמם משתמשים לצורך פעולתם. גם אלה משתמשים לכל דבר, ויש לפקח על ההרשאות שלהם. דוגמה לכך היא המשתמש IWAM_*ComputerName*, שנוצר על ידי IIS. את המילה *ComputerName* מחליף שם השרת בו הוא נוצר.

גם SQL Server יוצר לעצמו משתמשים, וחשוב מאוד לבדוק את הרשאותיהם.



טיפ:

המשתמשים Administrator ו-Guest הם יעדים ראשונים להתקפת פורצים. נבצע מספר פעולות שיקשו עליהם את החיים:

1. שנו את שמות החשבונות הללו לשמות אחרים, הידועים רק לכם.
2. למשתמש בעל הרשאות Admin תנו סיסמה קשה לפיצוח. בבקשה לא 1234 ! 6 ספרות ו/או אותיות לפחות.
3. צרו 2 חשבונות חדשים: Administrator ו- Guest. נעלו את החשבונות הללו (סמנו את התיבה Account Disabled). אלה הם פיתיונות לפורצים מתחילים...

קבוצות משתמשים – User Groups

כשמספר המשתמשים עולה לעשרות או מאות, נעשית המשימה של תחזוקת הזכויות וההרשאות של כל אחד מהם למתישה. כלי עזר חשוב למשימה זו הן קבוצות המשתמשים. מנהל המערכת מגדיר מספר קבוצות תפקוד בארגון: מנהלה, שיווק, פיתוח, מחסן וכדומה. לכל קבוצה הוא נותן זכויות גישה ושימוש במשאבי המערכת, ואז הטיפול במשתמש הבודד, הופך לפשוט הרבה יותר - יש רק לשייך אותו לקבוצה המתאימה. השיוך לקבוצות נעשה דרך לחצן Groups בחלון המשתמש.

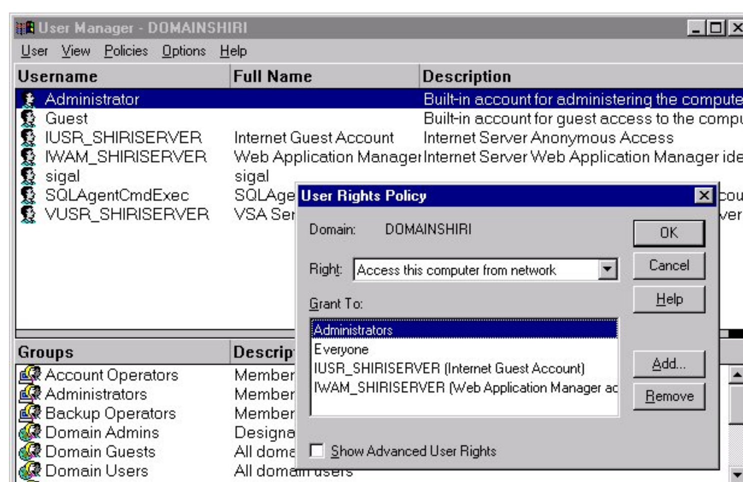
ישנן מספר קבוצות מוגדרות מראש:

תפקיד	קבוצה
מנהלי המערכת - בעלי גישה מלאה לכל משאבי המערכת.	Administrators
משתמשים - מותר להם לבצע פעילות רק לפי ההרשאות שלהם.	Users
אורחים - משתמשים ללא חשבון קבוע. זהו חשבון מסוכן שיש להגביל אותו ככל האפשר.	Guests

קבוצה	תפקיד
Server Operators	בעלי זכות לנהל את משאבי השרת.
Account Operators	בעלי זכות לניהול חשבונות: הוספת משתמשים וכדומה.
Print Operators	בעלי זכות לניהול המדפסות.
Backup Operators	בעלי זכות להפעיל גיבויים ושחזורים.

זכויות המשתמש

לכל משתמש ניתן לקבוע זכויות נפרדות. אל חלון הזכויות מגיעים מ- User Manager.

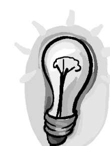


תרשים 22.2: User Rights Policy

בחלון זה תוכלו לקבוע עבור כל אפשרות את המשתמשים הזכאים לה. מדובר כאן על הרשאות רשת, ולא על זכויות גישה לקבצים בודדים. גם כאן כדאי לשים לב למשתמש האינטרנט - IUSER_*ComputerName*, ולהגביל אותו כך שלא יפריע לגולשים תמימים, אך לא ייתן סיכוי לפורץ.

ודאו שלמשתמש האינטרנט שלכם יש זכויות גישה (גם Read וגם Write) לקובץ מסד הנתונים שלכם, אחרת תקבלו את ההודעה המוכרת והלא חביבה:

Operation must use an updateable query



טיפ:

הסירו את הקבוצה Everyone מרשימת הזכאים ל:
Access this computer from network. בתרשים שלמעלה, סמנו את Everyone ואז לחצו על Remove.

גישה לקבצים

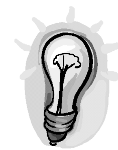
מערכת Windows NT יכולה לעבוד בכל אחד משני סוגים של מערכת קבצים:

FAT – השיטה הישנה שבה פועלות גם DOS וגם Windows 9x. פשוטה לתפעול, חסרת מנגנוני אבטחה לחלוטין. יתרון קטן: בעת תקלה ניתן להעלות את המחשב מדיסקט boot רגיל.

NT File System – **NTFS** - מערכת הקבצים של NT. מערכת חזקה בהרבה, מאובטחת ברמה גבוהה. ניתן לשלוט על רמת הגישה של משתמשים עד רמת קובץ בודד.

כדי לקבל אבטחה מקסימלית, כדאי לאתחל את כל הדיסקים בשרת בשיטת NTFS.

מנהלי רשת רבים נוהגים ליצור מחיצה (Partition) קטנה יחסית המפורמטת ב-FAT, עליה יושבת מערכת ההפעלה. את המחיצה השנייה מכינים ב-NTFS, ועליה מותקנות שאר התוכנות, כדוגמת IIS. היתרון הוא, שבעת תקלה ניתן להפעיל את המחשב מדיסקט. רעיון החלוקה למחיצות הוא רעיון מצוין. גם אם הצליח הפורץ להשתלט על מחיצת ה-IIS, ייתכן שלא יצליח לעבור למחיצת system, ולגרום נזקים חמורים יותר. אך העבודה עם FAT היא שיטה פסולה מבחינת אבטחה. מחיצת FAT היא פרצת אבטחה, וכדאי מאוד להימנע ממנה. עבור מקרים של תקלות, כדאי להשתמש במנגנוני גיבוי מסודרים, ובנושא זה אסור לחסוך. לסיכום: כדאי לבנות שתי מחיצות או יותר, כולן NTFS.



טיפ:

מערכת NTFS מסוגלת, לצורך תאימות אחורה, לייצר קבצים בפורמט 8.3. מכיון שתכונה זו פוגעת באבטחה, יש "לכבות" אותה. נתון זה נמצא ב-Registry ולכן הדרך לעשות זאת, היא דרך העורך regedit. אם מעולם לא הפעלתם עורך זה, עבדו בזהירות או בקשו עזרה. לחצו Start > Run, והקלידו regedit. הנתונים לשינוי הם:

Registry Hive:	HKEY_LOCAL_MACHINE\SYSTEM
Registry Key:	System\CurrentControlSet\Control\FileSystem
Name:	NtfsDisable8dot3NameCreation
Type:	REG_DWORD
New Value:	1

לאחר שווידאנו שכל הדיסקים שלנו עובדים ב-NTFS, ניתן לגשת למתן הרשאות גישה למשתמשים. על כל קובץ או תיקיה שעליהם נלחץ לחיצה ימנית, ונבחר Properties, נקבל את חלון מאפייני התיקיה או הקובץ. נבחר בכרטיסיה Security (אם לא מופיעה לכם כרטיסיה זו, כנראה מערכת הקבצים של דיסק זה אינה NTFS !).

בכרטיסיה זו מרוכזים שלושה היבטים של אבטחה:

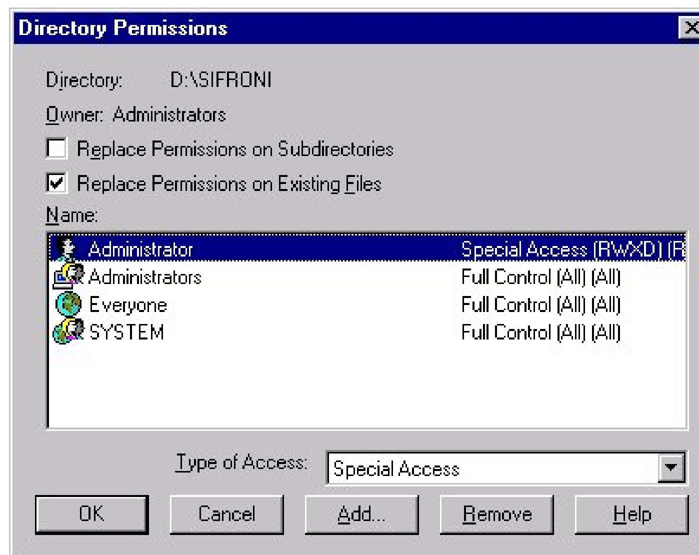


Permissions – כאן נקבע למי יהיו הרשאות גישה לקבצים השונים.

Auditing – מעקב אחר פעילות. מי ניגש לקובץ ומתי. על אילו קבצים ליצור מעקב?

Ownership – בעלות על קבצים.

נלחץ על Permissions ונקבל את חלון Directory Permissions. אנו רואים את המשתמשים או הקבוצות, ולכל אחד מהם יכולות להיות זכויות שונות. ניתן כמובן להוסיף משתמשים. בחלון ההוספה נקבע מה יהיו זכויותיו של המשתמש החדש לגבי משאב זה.



תרשים 22.3: חלון Directory Permissions

לסיום נושא האבטחה ב-Windows NT, הנה עוד כמה טיפים חשובים.

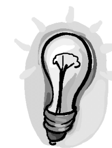


טיפ:

ישנן ב-NT מספר תוכניות ניהול. הגישה אליהן מקילה את הפריצה לשרת, ולכן נחסום גישה אליהן. אתרו את קבצי התוכניות הללו, והעבירו אותם למדור נפרד. למדור זה תנו הרשאות גישה רק למנהלי המערכת, קבוצת Administrators.

התוכניות הן:

ftp.exe, net.ext, cmd.exe, telnet.exe, wscript.exe, cscript.exe



טיפ:

מנעו גישת Anonymous לשרת שלכם. גם שינוי זה מתבצע דרך ה-Registry. מובן שאותם כללי זהירות שהוזכרו קודם, חלים גם כאן.

Registry Hive:	HKEY_LOCAL_MACHINE
Registry Key:	System\CurrentControlSet\Control\LSA
Name:	RestrictAnonymous
Type:	REG_DWORD
New Value:	1



טיפ:

הסירו את שירותי הרשת שאינם דרושים לכם. כמה מהם עלולים להוות פרצות אבטחה:

Alerter

Clipbook server

Computer Browser

DHCP Client

Messenger

NetBios interface

NetLogon

Scheduler

Server Service

Simple TCP/IP Services

Spooler

TCP/IP NetBios Helper

Wins Client (TCP/IP)

Workstation Service



טיפ:

הסירו את שם המשתמש האחרון מתיבת ה Log On. בדרך כלל שם המשתמש האחרון מופיע בתיבה זו, וכל מה שנותר לנחש הוא הסיסמה. אם גם שם המשתמש לא יהיה ידוע, קשה יותר יהיה לפרוץ לשרת שלכם.

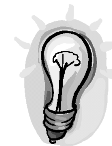
Registry Hive: HKEY_LOCAL_MACHINE

Registry Key: software\microsoft\windows NT\current version\winlogon

Name: DontDisplayLastUserName

Type: REG_SZ

New Value: 1



טיפ:

הסירו את הלחצן Shut Down מתיבת Log On. משמעותו של לחצן זה היא שניתן לכבות את השרת גם מבלי לדעת את השם והסיסמה. אנו כמובן רוצים למנוע זאת.

Registry Hive: HKEY_LOCAL_MACHINE

Registry Key: software\microsoft\windows NT\current version\winlogon

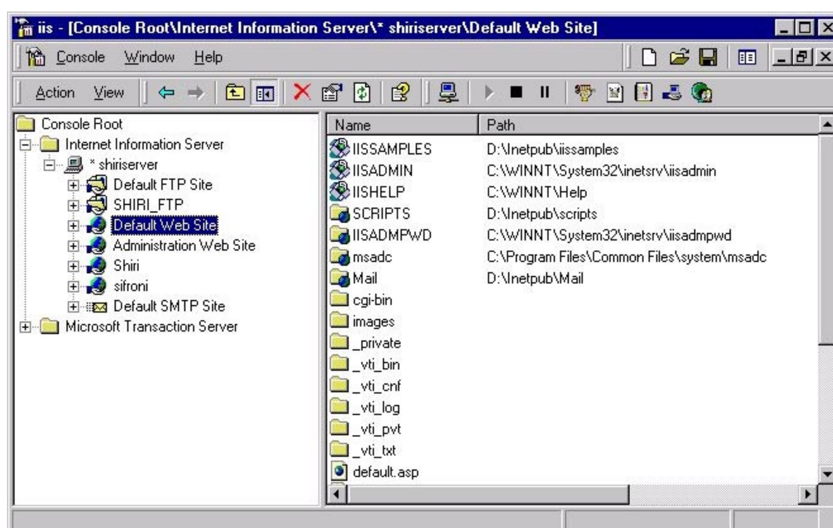
Name: ShutdownWithoutLogon

Type: REG_SZ

New Value: 0

הגנה על השרת – אבטחה ב- IIS

הכלי המרכזי לניהול IIS, הוא Internet Service Manager. כפי שנראה בתרשים, יכול השרת להכיל אתרים רבים, ולכל אחד מהם ניתן לקבוע את תכונותיו בנפרד.





טיפ:

הסירו את כל אתרי הדוגמה המותקנים על ידי תוכנית ההתקנה של IIS. להלן שמותיהם ומיקומם, אם לא שיניתם את ברירות המחדל.

IIS	C:\inetpub\iissamples
IIS SDK	C:\inetpub\iissamples\sdk
Admin Scripts	C:\inetpub\adminscripts
Data Access	C:\program files\commonfiles\system\msadc\samples



טיפ:

הסירו את התיקיה הווירטואלית (Virtual Directory) ISADMPWD.
תיקיה זו מאפשרת איפוס סיסמאות, והיא מהווה פרצת אבטחה.

לחצו לחיצה ימנית על אחד האתרים המאוחסנים בשרת, ובחרו Properties.

כאן ניתן לקבוע את שם האתר (שם פנימי בלבד, זה אינו בהכרח השם בו יופיע האתר באינטרנט!), ולבחור את כתובת IP שלו מתוך הכתובות שהוקצו לשרת.

sifroni Properties

Documents | Directory Security | HTTP Headers | Custom Errors
Web Site | Operators | Performance | ISAPI Filters | Home Directory

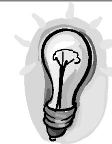
Web Site Identification
Description: sifroni
IP Address: 192.117.145.5 | Advanced...
TCP Port: 80 | SSL Port:

Connections
Unlimited
Limited To: 1,000 connections
Connection Timeout: 900 seconds

Enable Logging
Active log format:
W3C Extended Log File Format | Properties...

OK | Cancel | Apply | Help

האם לאפשר רישום ב-Log? בדרך כלל לא. הרישום גוזל זמן ומשאבי שרת. הסרת הרישום תשפר את ביצועי השרת.



טיפ:

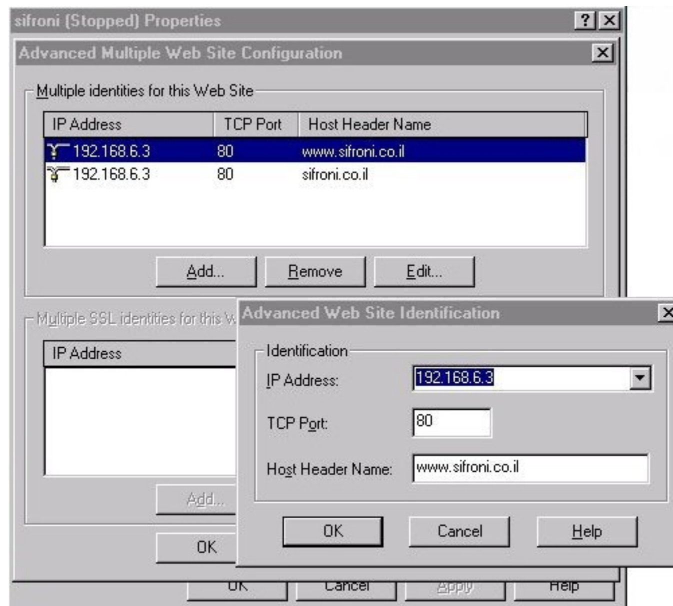
בחבילה NT Option Pack, כלול מוצר חמוד שנקרא Site Server Express. זו תוכנה למעקב סטטיסטי אחר האתר. תוכלו לקבל מתוכנה זו דוחות מפורטים על כמות המבקרים באתר, כפונקציה של שעות היום, של היום בשבוע וכן הלאה. התוכנה מפיקה גרפים יפהפיים, ויש לה אפשרויות רבות ומגוונות. את הדוחות המעודכנים ניתן לראות דרך האינטרנט, מפני שהם בעצם דפי HTML. כלומר, אתם או הלקוח שלכם יכולים לראות דוחות מעודכנים גם מהבית או מכל מקום אחר.

למה אני מזכיר את Site Server Express דווקא כאן? מפני שהוא מקבל את הנתונים לפעולתו מקובץ Log. כלומר, אם אתם משתמשים
ב- Site Server Express, סמנו את התיבה Enable Logging ובחרו בפורמט W3C. אם לא – בטלו את הסימון, חבל על הזמן.

טיפ:



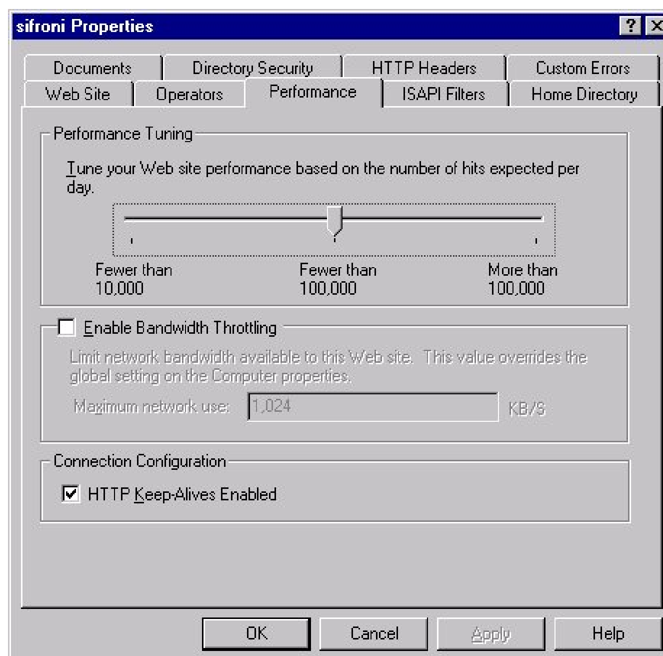
כיצד מאחסנים אתרים רבים על אותה כתובת IP? אם בניתם אתרים אחדים, לעצמכם או ללקוחותיכם, ונגמרו לכם הכתובות, כדאי לדעת שאפשר לאחסן אתרים רבים על אותה כתובת. צרו את האתרים כרגיל ב-IIS ותנו לכולם את אותה כתובת. ליד הכתובת יש לחצן בשם Advanced. לחצו עליו, ויתקבל החלון הבא:



בחלון זה עליכם להקליד את השיוך בין שם האתר לבין הכתובת. הקלידו כמו בדוגמה את הפרטים בכל האתרים, ואשרו. זה עובד!

כוונון ביצועים - Performance

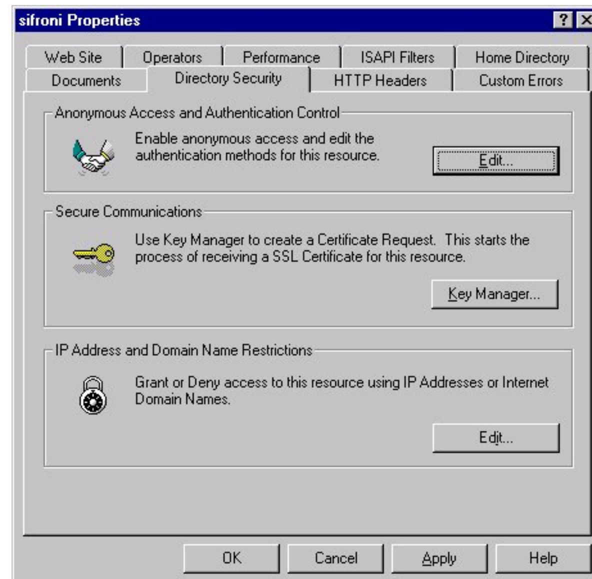
בכרטיסיה זו ניתן לכוון את ביצועי השרת באופן אופטימלי, לפי מספר המבקרים הצפוי. בזמן פיתוח יש לכוון את הגררה (slider) לכמות הנמוכה ביותר של המשתמשים, זה ייתן ביצועים מירביים. לאחר מכן, עקבו אחר כמות הגולשים היומית, וכוונו בהתאם.



האפשרות Enable Bandwidth Throttling מיועדת להגביל את רוחב הפס הניתן לאתר. אתם שואלים: מדוע שנרצה להגביל? נסביר זאת בעזרת דוגמה. לעיתים, יושבים בשרת שלכם מספר אתרים של לקוחות שונים. כל לקוח משלם לכם סכום קבוע עבור אחסנת האתר, נניח $100 לחודש. רוב האתרים מתנהגים יפה, אבל אחד מהם מציע לגולשים להוריד סרטי וידאו כבדים, או לשמוע מוסיקה בזמן אמת, או כל פעילות אחרת, שגוזלת רוחב פס רב. מה קורה במקרה זה? אתר זה גוזל את רוב רוחב הפס שרכשתם, והאחרים מקבלים ביצועים נחותים בגללו, והם מתלוננים. מה שצריך לעשות, הוא להגביל את רוחב הפס המוקצה לאתר הסורר, כך שלא יפריע לשכניו.

מכאן נובע, שכאשר אתם מאחסנים את אתרכם בחוות שרתים כלשהי, אל תסתפקו בפרטים כלליים כמו מחיר לחודש ונפח האחסון המוקצה לכם, אלא שאלו מהו רוחב הפס המובטח לכם. אם התשובה תהיה "אנחנו מקושרים לאינטרנט ב- 10MB/s!", פסלו זאת על הסף. לא מעניין אֶתכם מה רוחב הפס הכללי שרכשה החברה, אלא איזו פרוסה מתוכו מובטחת לכם!

Directory Security - אבטחת תיקיה



בכרטיסיה זו ניתן לשנות את קביעות האבטחה של התיקיה. היא מחולקת לשלושה חלקים, כן כמו כל דבר:

1. **Anonymous Access and Authentication Control**

אופן זיהוי הגולש. על כך בהמשך.

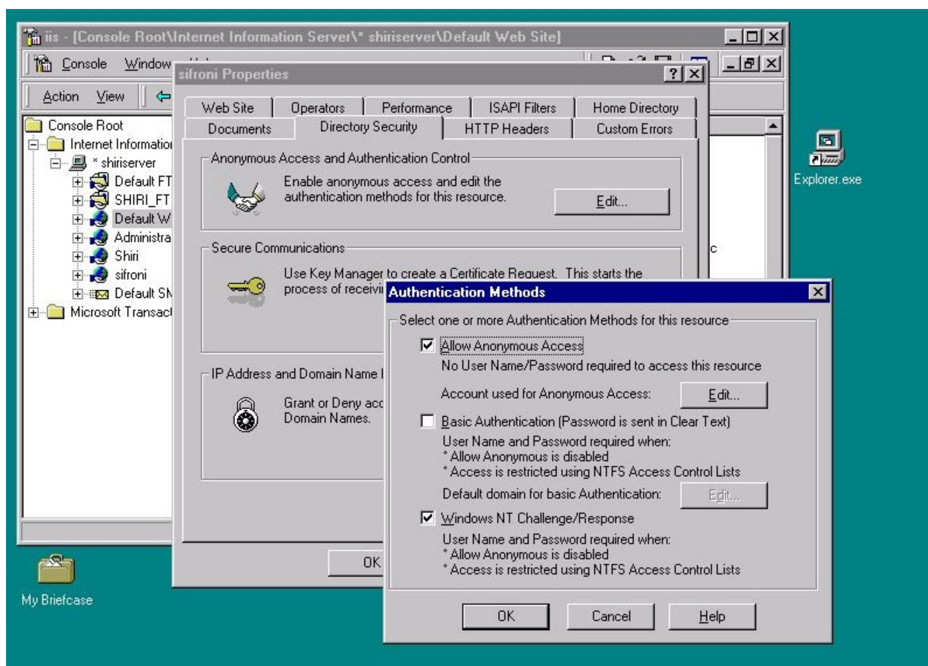
2. **Secure Communications**

אבטחת התקשורת. מפתחות, שיטות הצפנה, ועוד. הרחבה על כך בפרק הבא – אבטחה באמצעות SSL.

3. **IP Address and Domain Name Restrictions**

הגבלת גישה לפי כתובות IP ו-Domain Names.

בחלק הראשון, **Anonymous Access**, נלחץ על Edit ונקבל את חלון הזיהוי. התהליך של קביעת זהותו של הגולש נקרא זיהוי, Authentication. בחלון שיטות הזיהוי קיימות 3 שיטות זיהוי. תיבות הסימון אינן תלויות זו בזו, ניתן לסמן יותר משיטה אחת.



Anonymous

כדי שניתן יהיה להפעיל על גולש האינטרנט את מנגנון האבטחה החזק של Windows NT, חייב הגולש להיות מזוהה עם חשבון משתמש כלשהו. לשם כך הומצא חשבון מיוחד לגולשים, ושמו IUSR_*sysname*, כאשר את מקום המילה *sysname* תופס שם השרת. אם תעברו מספר דפים אחורה, אל תמונת User Manager, תוכלו לראות את חשבון משתמש האינטרנט (IUSR_*sysname*) בשרת שלי. אין צורך ליצור אותו, הוא נוצר אוטומטית בעת ההתקנה של IIS, וניתנת לו סיסמה אקראית. בהיותו משתמש רגיל, ניתן לערוך את זכויותיו של IUSR_*sysname*, ולהגביל אותו במקומות הנכונים.

אם סומנה התיבה Anonymous – תורשה כניסתו של הגולש האנונימי.

Basic Authentication

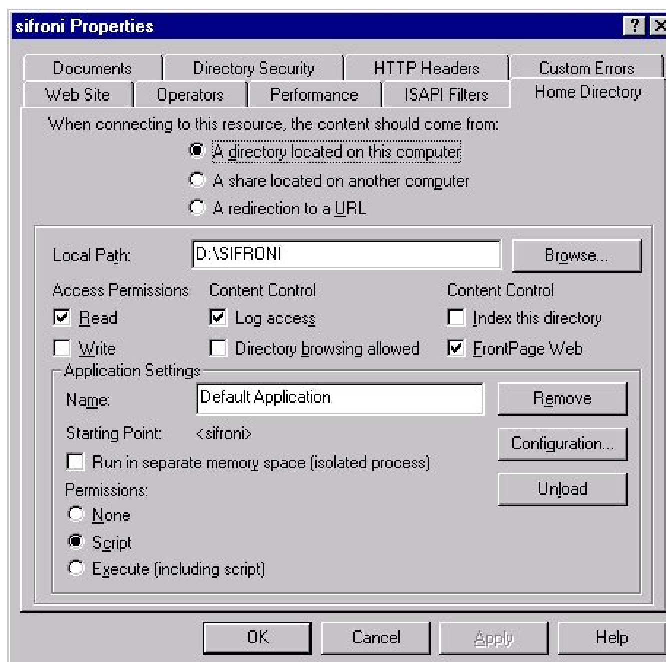
אם ביקש הגולש IUSR_*sysname* משאב שהגישה אליו אסורה לו, נשלחת הודעה 401 (שפירושה: אין גישה) לדפדפן הגולש. אפשרות זו סומנה, ואם הדפדפן תומך בכך, תוצג בפני הגולש תיבת שיחה המבקשת שם וסיסמת גישה למשאב. הגולש מקליד שם וסיסמה, ואלה נבדקים בשרת. אם אכן יש משתמש כזה, הופך הגולש להיות משתמש זה, ולא עוד גולש אנונימי.

Challenge/Response Authentication

שיטת זיהוי זו דומה מאוד לשיטה הקודמת, בהבדל קטן: בעוד שבשיטת הזיהוי הבסיסי נשלחים השם והסיסמה כמו שהם, כאן הם עוברים הצפנה.

כרגע רק Internet Explorer תומך בשיטת זיהוי זו.

מיקום פיסי של קבצי האתר - Home Directory



בכרטיסיה זו תוכלו לקבוע מספר פרמטרים חשובים, הקשורים למיקום הפיסי של קבצי האתר.

החלטה ראשונה: מהיכן יבוא תוכן האתר?

- מתיקיה במחשב זה (זו בדרך כלל האפשרות הנבחרת).
- מתיקיה משותפת במחשב אחר ברשת המקומית.
- הפניה לכתובת אינטרנט אחרת. אם תבחרו באפשרות זו – תתבקשו לציין את הכתובת.

Local Path: זוהי התיקיה בדיסק בו נמצאים קבצי האתר. ביחס אליו, תוכלו לפתוח תיקיות נוספות כמו images ואחרים. לחצן browse מקל על העבודה.

הרשאות גישה - Access Permissions

האפשרות Read מאפשרת קריאת קבצים, ולכן הכרחית.

האפשרות Write מאפשרת כתיבה לתיקיה זו. ההחלטה על סימון תיבה זו צריכה להיעשות לפי פעילות האתר. האם המשתמשים אמורים לשלוח אלינו קבצים, או לשנות קבצים כלשהם? אם כן – סמנו. אם לא - כדאי לא לסמן, ולמנוע בכך אפשרות של נזק בזדון.

Log Access – רישום פעילות הגולש בקובץ Log. גם כאן, הרישום גוזל משאבים, לכן לא נבקש אותו, אם לא משתמשים במידע.

Directory Browsing Allowed. האם יוכל המשתמש לראות את מבנה התיקיות של האתר? בדרך כלל, התשובה היא חד-משמעית לא. עם זאת, כדי לאפשר ניפוי שגיאות (Debugging) יעיל, יש לסמן אפשרות זו. לא לשכוח למחוק אפשרות זו כאשר האתר הופך פעיל.

Application Settings

כאן נקבעת רמת האבטחה הבסיסית של התיקיה המסומנת. שלוש רמות אפשריות:

None – שום קובץ הנמצא בתיקיה זו לא יכול להיות מופעל. זו רמת האבטחה הגבוהה ביותר, אך בדרך כלל לא נשתמש בה, מפני שהיא מונעת את דפי ASP שלנו מלפעול.

Script – רק דפי קוד, כדוגמת ASP, יכולים להיות מופעלים בתיקיה זו. זו רמת האבטחה המקובלת באתרים מסחריים.

Execute – כל תוכנית או יישום (קבצי asp) הנמצאים בתיקיה זו יכולים לפעול. זו רמת האבטחה הנמוכה ביותר. נבחר בה רק אם חייבות להיות תוכניות הפעלה כלשהן בתיקיה. בדרך כלל נעדיף להעביר אותן לתיקיה אחרת ורק בה להרשות הפעלה. בתיקיה בה נמצאים קבצי היישום שלנו, לא נסמן אפשרות זו.

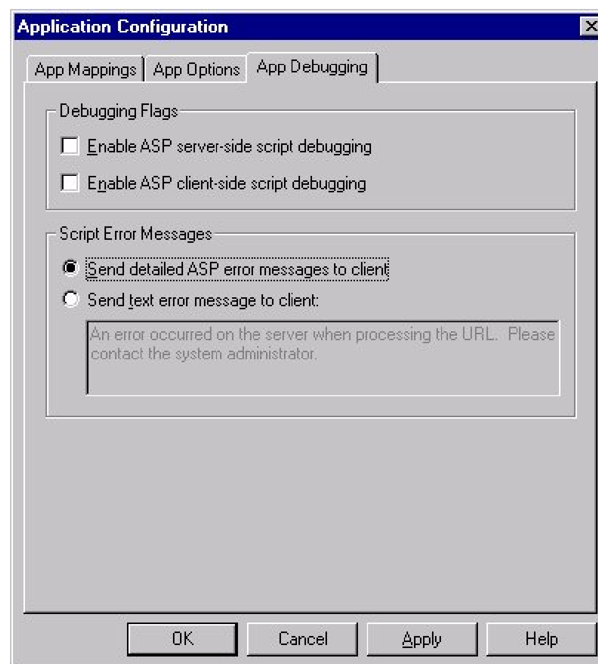
לחיצה על הלחצן Configuration תביא חלון נוסף וחשוב למפתחי ASP. חלון Application Configuration מאפשר לנו לקבוע כמה תכונות חשובות של המערכת.

Application Configuration

לחיצה על לחצן Configuration מביאה אותנו אל חלון Configuration שבו 3 כרטיסיות, בהן ניתן לקבוע מספר אפשרויות לגבי היישום. נסקור את החשובות שבהן.

App Debugging

האם ברצונכם לאפשר Debugging? בדרך כלל, במהלך הפיתוח, תהיה התשובה חיובית. סמנו את שתי תיבות הסימון ואפשרו Debugging הן בצד הלקוח והן בצד השרת. אל תשכחו לבטל תיבות אלה כאשר האתר נפתח לקהל.



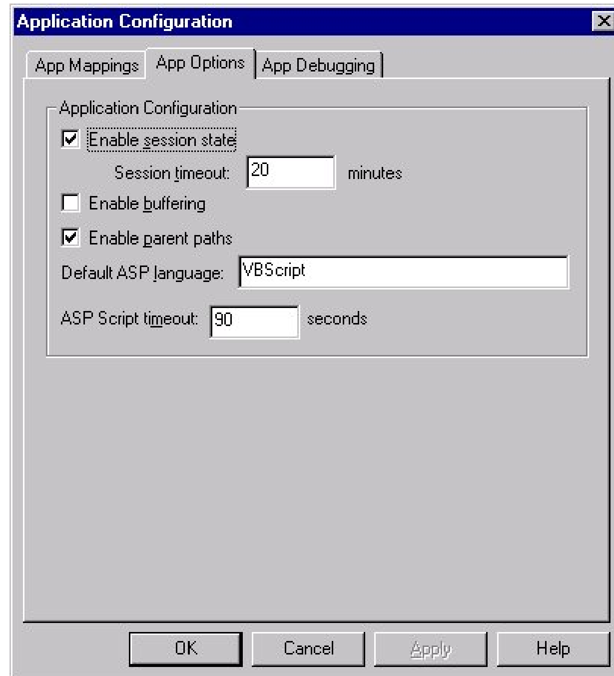
הודעות שגיאה

מה יקרה כאשר מתגלה שגיאה בדף ASP? יש שתי אפשרויות: האחת, הודעת שגיאה מפורטת נשלחת ללקוח. זו האפשרות הנכונה לכל שלבי הפיתוח, אחרת לא נדע מה השגיאה, וקשה מאוד יהיה לתקן אותה. אבל כאשר האתר נפתח לקהל הרחב, אולי לא תרצו שהגולש בבית יקבל מידע מדויק על מהות התקלה. זה לא נראה טוב, ויש בכך גם סיכון. במקרה זה סמנו את האפשרות השנייה – שלח הודעה קבועה. כתבו במקום המיועד לכך את הודעתכם.

App Options

כאן ניתן לכוון מספר אפשרויות הקשורות לאפליקציות ASP.

לדוגמה, קבוע הזמן של ה-Session, הזכור לנו מהפרק בו בנינו מוני מבקרים. לאחר זמן זה – ללא תגובה מהמשתמש – מסתיים ה-Session.



קיר אש - FireWall

קיר אש הוא מוצר המיועד להגנה על הרשת הפנימית בארגון מפני חדירות עוינות מהאינטרנט. בדרך כלל מדובר במוצר תוכנה. כלומר: מחשב רגיל המוקדש למטרה זו, ועליו מותקנת תוכנת FireWall. יש לו שני כרטיסי רשת (ולעיתים יותר) – האחד מחובר לאינטרנט, והשני לרשת הארגונית. במקרים אחרים מגיע המוצר כמכשיר סגור: קופסה מוכנה עם שקעי תקשורת: האחד לאינטרנט, והשני לרשת.

בשני המקרים תפקיד המערכת זהה: להיות שוטר תנועה ופקח גבולות של התקשורת העוברת בין החברה לעולם האינטרנט.

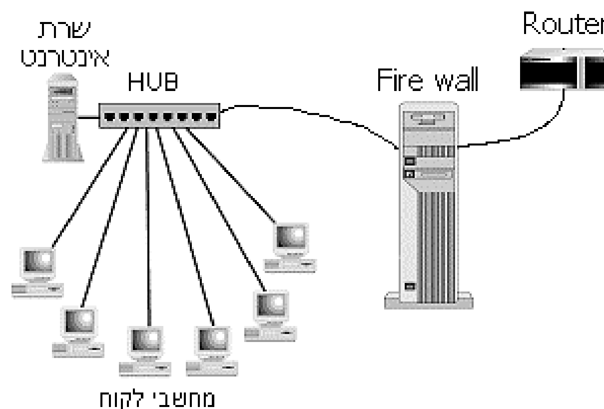
יחידת התקשורת הבסיסית בפרוטוקול TCP/IP היא ה-Packet, מנה/חבילה. ממש כמו חבילה בדואר, גם על חבילת TCP/IP רשום למי היא מיועדת, וממי נשלחה. כל התקשורת ברשת האינטרנט מורכבת ממנות קטנות כאלה. כל דפי האינטרנט שאתם מורידים, כל התמונות, כל הדואר האלקטרוני, הכל מורכב ממנות. בדרך כלל לא תספיק מנה אחת להכיל את כל הקובץ שברצונכם להעביר, ולכן מתבצעת ההעברה בעזרת מנות רבות וממוספרות. רק כשהגיעו כולן ליעדן בשלום – נבנה הקובץ הסופי

ומוצג לפניכם. בנוסף לכתובת השולח והנמען, יש למנה גם נתון נוסף הנקרא Port. ה-Port הוא מספר המסמן את סוג השירות. לכל סוג שירות יש מספר קבוע. דוגמאות:

Port	שירות
21	FTP – File Transfer Protocol
23	Telnet – Terminal Emulation
25	SMTP – Simple Mail Transfer Protocol
80	HTTP – HyperText Transfer Protocol
443	HTTPS – Secure HTTP

כך ניתן להבחין בין מנות השייכות לשירותים השונים.

מבנה כללי של מערכת מוגנת FireWall



למערכת FireWall שני צדדים: חיצוני, הפונה לעולם, ופנימי הפונה לרשת הארגונית. המערכת בודקת כל חבילה שעוברת בין שני הצדדים ומחליטה אם להעביר אותה או לא. הנחת העבודה הבסיסית היא שהכל אסור. רק מה שהחלטתם במפורש להעביר – יעבור. עליכם לקבוע כללים, Rules. כלל אומר: מנה שתגיע מכתובת A לכתובת B, ב-Port C – יש לבצע עליה פעולה D .

דוגמה: מנה המגיעה מכל כתובות הרשת הפנימית לכתובתו של ספק האינטרנט שלנו, ב-Port 80 – העבר אותה. המשמעות של כלל זה: הרשה לעובדי החברה לגלוש באינטרנט.

באמצעות שליטה מדויקת בחוקי התעבורה של הרשת, ניתן להשיג רמה גבוהה מאוד של אבטחה, וכמה יתרונות נוספים:

VPN – Virtual Private Network - נניח שיש לחברה שלנו סניפים ברחבי העולם. באמצעות הגדרה מתוחכמת של FireWall ניתן ליצור רשת פרטית משותפת ומאובטחת שכל הסניפים מחוברים אליה, על גבי רשת האינטרנט. כדי להשיג קישוריות כזו עד כה היתה החברה צריכה לרכוש קווי נל"ן יקרים ביותר.

שימוש בכתובות מזויפות - כתובות IP הן כידוע מצרך מבוקש מאוד. הכתובות פשוט הולכות ונגמרות, וצרכני הכתובות - גולשי אינטרנט ואתרי אינטרנט – הולכים ומתרבים. עד לא מזמן קיבלה כל חברה שרצתה להתחבר לאינטרנט, כתובת אמיתית לכל עובד. כעת, בעזרת טריק שנקרא NAT – Network Address Translation - תרגום כתובות רשת, יכולה חברה גדולה לחבר מאות עובדים לרשת האינטרנט בעזרת כתובת IP בודדת. תוכנת FireWall שומרת לעצמה טבלת המרה בין כתובות IP "מזויפות" לבין שמות המחשבים ברשת הפנימית. אין צורך להיבהל מהמילה "מזויפות", אין כאן שום פשע. אלה פשוט כתובות שלא הוקצו לכם על ידי ארגון האינטרנט העולמי, ולכן אסור לכם "לצאת" איתן לעולם, אלא רק לעשות בהן שימוש פנימי.

מתן כתובת מזויפת לשרת האינטרנט שלכם הוא טכניקה מצוינת לאבטחת השרת. כך מציג עצמו השרת שלכם בעולם בכתובת החוקית, אך נמצא למעשה בכתובת אחרת. FireWall מבצע כל הזמן תרגום (מבוקר!) בין הכתובות. מי שינסה לתקוף את השרת שלכם, יפנה את מאמציו לכיוון הכתובת הידועה, החוקית, אך לא ימצא שם דבר מלבד ה-FireWall שהוא מטבעו מכונה מוגנת מאוד.

שימוש ב-FireWall עולה בביצועים. המרת הכתובות ובדיקת המנות העוברות בין צידי ה-FireWall לוקחות זמן, ולכן יש בכך פגיעה מסוימת בביצועים. הירידה בביצועים, עם זאת, אינה גדולה. השיפור באבטחה לעומת זאת הוא עצום, ולכן כדאי בכל מקרה שאפשר, להשתמש בתוכנת FireWall כלשהי.

פרק 24

הגנה באמצעות פרוטוקול SSL

שיטות הצפנה

מהי הצפנה? הצפנה היא תהליך, בו משנים מידע כלשהו באמצעות הפעלת תהליך ידוע על המידע, לצורך הסתרתו מגורמים עוינים. מי שיודע את התהליך שהופעל, יכול להפעיל תהליך הפוך ולקבל את המידע בקלות. לכל השאר, פיענוח המידע הוא קשה, גוזל זמן רב, ולמעשה בלתי אפשרי.

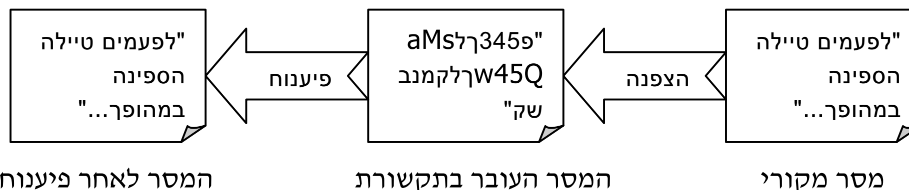
ברוב שיטות ההצפנה, התהליך עצמו ידוע ומפורסם, אך הוא משלב נתון מסוים, מספרי, או טקסטואלי, הנקרא "מפתח". המפתח משולב בתהליך ההצפנה, והכרחי לתהליך הפיענוח. ממש כמו מפתחות הבתים שלנו: כולם בנויים באותה תבנית, אך יש מספר, קוד, צופן, שמאפיין כל אחד מהם, ורק בעזרת הקוד הנכון, ניתן לפתוח את הדלת.

בתחום אבטחת המידע, ממלאים אלגוריתמים מתמטיים את תפקיד מנעולי הפלדה. פונקציה מתמטית מופעלת על המידע, ומערבלת אותו, כך שקשה יהיה לפענח אותו. מאחר ופונקציות הערבול ידועות, נותר למפצחים רק לנחש את מספר המפתח. בעידן המחשב יש לכך פתרון: ניתן למחשב לנסות את כל האפשרויות! המחשבים כידוע, הולכים ונעשים רבי עוצמה, ובדרך כלל, זו משימה פתירה.

כאן נכנס למשחק אורך המפתח. והנה לכם תחום, שבו האורך בהחלט קובע. עד כה היה מקובל להשתמש במפתחות בני 40 Bit. ייצוא מארה"ב של טכנולוגיות הצפנה עם מפתחות ארוכים יותר – נאסר! אך כיום הסתבר, שמחשב פנטיום מהסוג שנמצא בכל בית, מסוגל לפצח קוד 40 Bit בכמה שעות, פשוט על ידי בדיקת כל האפשרויות הקיימות. כל Bit נוסף, מכפיל ב-2 את כמות האפשרויות. לכן התקבל התקן חדש, הגורס מפתחות בני 128 Bit. זהו מפתח המספק מספר עצום של אפשרויות, והוא המקובל כיום באבטחת העברות כספים וסחר אלקטרוני.

הצפנה סימטרית

בשיטת הצפנה סימטרית משתמשים שני הצדדים באותו מפתח. המפתח שהצפין – הוא בדיוק זה שיפענח. מובן שזה צריך להיות מספר סודי, שרק שני הצדדים יודעים אותו.



זוהי שיטה פשוטה ויעילה להעברת מידע מוצפן בין שתי נקודות. היא מצוינת לשימוש בין שני אנשים או גופים, שמסכמים ביניהם על מפתח, ורק הם יודעים אותו. ברשת האינטרנט זוהי שיטה בעייתית. אנו רוצים ליצור עם הגולשים תקשורת מאובטחת, כיצד נודיע להם מהו המפתח? חייבים להעביר אותו בתקשורת בלתי מוצפנת, כי עוד לא הקמנו קשר מוצפן! ואז יכול כל אחד "לעלות" על המפתח. מעבר לכך: כיצד נבדיל בין גולש תמים לבין האקר, שינצל את ידיעת המפתח לפיענוח הודעות של אחרים? ובכלל, כיצד אפשר לסמוך על גולש אינטרנט, שלא יעביר את הקוד הסודי לאחרים? אכן מגבלות קשות, ויש לפתור אותן. אך יש גם צד חיובי: הצפנה סימטרית דורשת מעט משאבי מחשב, יחסית, והיא פשוטה למימוש. כלומר, היא תאט את המחשב פחות משיטות אחרות.

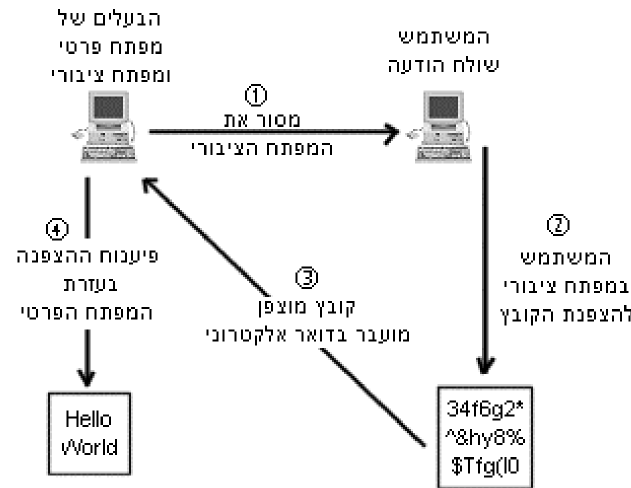
הצפנה אסימטרית

בשיטת ההצפנה האסימטרית, יש לכל צד שני מפתחות: אחד נקרא "מפתח פרטי", שלעולם אסור לגלות אותו, השני נקרא "מפתח ציבורי", שמותר לגלות אותו לכל המעוניין.

וזהו הטריק: מסר שהוצפן במפתח אחד, יכול להיות מפוענח רק בעזרת המפתח השני!

כל צד שומר בקפידה על המפתח הפרטי שלו, ומוכן לגלות לצד השני רק את המפתח הציבורי. כאשר צד א' רוצה לקבל מידע סודי מצד ב', הוא שולח לו את המפתח הציבורי שלו, ומבקש את המידע. צד ב' מכין את המידע הדרוש, מצפין אותו בעזרת המפתח הציבורי של א' אותו קיבל, ושולח לצד א'. כל הפעולות הללו נעשות בערוץ לא מוגן, אך רק צד א', שבידו המפתח הפרטי, "בן הזוג" של המפתח בו הוצפן המידע, יוכל לפענח את ההודעה.

נדגים זאת בתרשים הבא:



הצפנה ופיענוח בעזרת מפתח פרטי ומפתח ציבורי

למחשב השמאלי יש מפתח פרטי ומפתח ציבורי. את המפתח הציבורי הוא שולח למחשב הימני (שלב 1). בעזרת המפתח הציבורי, מצפין המחשב הימני את המסר (שלב 2) ושולח אותו למחשב השמאלי (שלב 3). רק המחשב השמאלי יכול לפענח את המסר שהוצפן בעזרת המפתח הציבורי שלו וזאת על ידי שימוש במפתח הפרטי שברשותו (שלב 4).

חתימה דיגיטלית

חלק חשוב מאבטחת התקשורת, היא בדיקת המקוריות של המסר. כלומר: מה שהגיע אלינו הוא בדיוק מה שנשלח, ולא עבר שינויים ותוספות בדרך. כשמדובר במפתחות אסימטריים, הבעיה אינה קיימת, מפני שלא ניתן יהיה לפענח הודעה, שעברה טיפול כלשהו. המפתחות מתוכננים כך, ששינוי כלשהו במסר המוצפן, יגרום לכך שהמקבל לא יוכל לפענח את המסר כלל. אבל לעיתים, רוצים לפרסם מסמך שאינו מוגן, אך רוצים בכל זאת לתת בידי המקבל כלי לבדיקה ואישור, שזהו אכן המסמך המקורי, ללא תוספות וללא השמטות. לשם כך, פותחו החתימות הדיגיטליות. זוהי בעצם פונקציה מתמטית, הפועלת על המסמך, ותוצאתה קצרה בהרבה מן המסמך. אי אפשר לפענח את המסמך המקורי מתוך החתימה. החתימה הדיגיטלית היא "טביעת האצבע" של המסמך המקורי, והיא אופיינית רק לו. פונקציות חישוב החתימה, הנקראות לעיתים Hash Algorithms, מתוכננות כך, שיהיה זה קשה מאוד, ומעשית בלתי אפשרי, לשנות את המסמך כך שיהיה שונה, אך ייתן את אותה חתימה דיגיטלית.

יצירת חתימה דיגיטלית אינה משנה את המסמך. החתימה הדיגיטלית היא קובץ נפרד. בעת קבלת המסמך והחתימה, המקבל מפעיל על המסמך את אותו אלגוריתם, ואז הוא יכול להשוות את תוצאת החישוב עם החתימה שקיבל. אם יש זהות – המסמך המקורי.

רשויות רישוי

ראינו כי על פי החתימה הדיגיטלית, יכול אדם לוודא, שהאדם העומד מולו הוא אכן מי שהוא טוען שהוא. אבל מי יכול למנוע מכל אדם לייצר חתימות דיגיטליות ומפתחות פרטיים וציבוריים, ולהפיצם באינטרנט? לשם כך, יש רשויות רישוי - Certificate Authorities, ובקיצור CA. אלה הן חברות, שהוסמכו לכך, ורק להן מותר להנפיק מפתחות. דוגמאות לחברות כאלה:

Verisign	www.verisign.com
Entrust Technologies	www.entrust.com
Thawte	www.thawte.com
EauroSign	www.eaurosign.com
Xcert	www.xcert.com

מומלץ לבקר באתרים של חברות אלה, יש בהן מידע רב על אבטחה בכלל ועל צורת העבודה מולן בפרט. לאתרים מסחריים נמכרים המפתחות בתשלום. בחלק מה-CA ניתן להוריד גם "מפתח דמו", לניסיון למספר שבועות. כאשר תבקשו לרכוש לעצמכם מפתח, תשאל אֶתכם רשות הרישוי שאלות רבות, ואמיתות התשובות תיבדק. לאחר שתעמדו בכל הדרישות – תקבלו "רשיון דיגיטלי".

מהו רשיון דיגיטלי?

רשיון דיגיטלי הוא רשומה ייחודית, המכילה מידע הן על בעל הרשיון (חברה, גוף, או אדם), והן על רשות הרישוי - ה-CA שהפיקה אותה. התקן הנפוץ כיום לרשיונות דיגיטליים הוא X509.

לפי תקן X509, חייב הרשיון להכיל את המידע הבא:

- גרסת הרשיון.
- מספר סידורי של הרשיון.
- סוג האלגוריתם, ששימש לחתימת הרשיון.
- שם רשות הרישוי, שהפיקה את הרשיון.
- תאריך תחילת תוקף ותאריך סיום תוקף הרשיון.

* שם בעל הרשיון.
* המפתח הציבורי של בעל הרשיון.
* החתימה הדיגיטלית של רשות הרישוי, שהפיקה את הרשיון.

בדרך כלל, הרשיון יכיל גם פרטים נוספים, כגון: כתובת דואר של בעל הרשיון, מדינה, כתובת E-mail, תחום עיסוק וכן הלאה.

שני פרטים חשובים ביותר ברשיון, הם שם ה-CA, וחתימתו הדיגיטלית. על פי שני אלה, יכול מקבל ההודעה לוודא, בעזרת רשות הרישוי (CA), ששולח ההודעה הוא אכן מי שהוא טוען שהוא. זהו חלק חשוב בתפקידו של ה-CA, להיות ה"נוטריון" של עולם האינטרנט.

פרוטוקול SSL

Secured Sockets Layer, הוא כנראה הפרוטוקול הנפוץ ביותר כיום להעברת נתונים מאובטחת ברשת האינטרנט, שפותח על ידי Netscape במיוחד למטרה זו. SSL הוא פרוטוקול עצמאי, שאינו תלוי ביישום, ומכאן שהוא יכול לעבוד תחת http, ftp ואפילו telnet.

קל מאוד לדעת אם אתם באתר מאובטח ב-SSL, לפי התחילית https, במקום http כרגיל.

SSL משתמש בהצפנה, באמצעות מפתחות ציבוריים (Public key Cryptography), כדי לוודא את זהות השרת, עוד לפני העברת המידע ברמת היישום.

כיצד עובד פרוטוקול SSL?

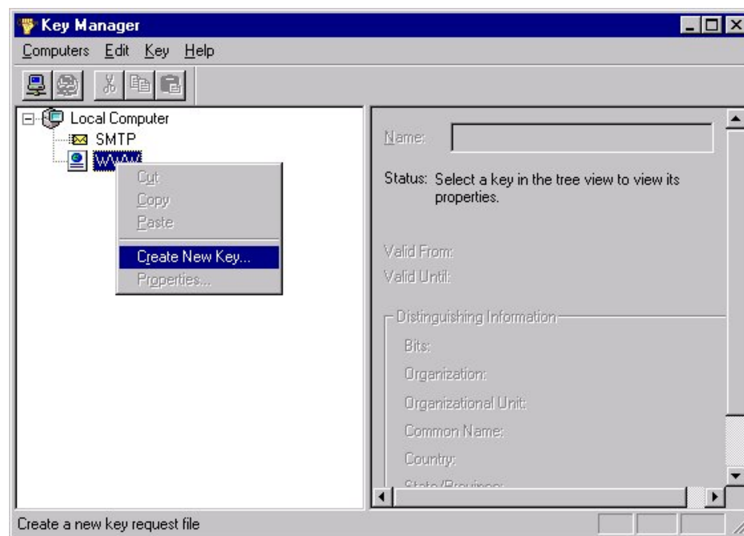
תקשורת בין שני גופים בפרוטוקול SSL, מתחילה בשלב ראשון, שנקרא Handshake – לחיצת יד. בשלב זה מחליפים השרת והלקוח מפתחות ביניהם, וכל אחד מוודא שהצד השני הוא אכן מי שהוא טוען שהוא. שלב זה חייב להיות מבוצע בהצפנה **אסימטרית**, מפני שאי אפשר להעביר מפתח סודי בערוץ בלתי מוגן. בסיום שלב Handshake, מייצר הלקוח קוד סודי, ומוסר אותו לשרת. מאותו רגע יש לשני הצדדים קוד סודי, הידוע רק לשניהם, ואז הם עוברים להצפנה סימטרית, שהיא כזכור קלה הרבה יותר למימוש, ומעמיסה פחות על ה-CPU. בהצפנה סימטרית, מבוצעת כל הפעילות העסקית, שלכבודה היתה כל הטרחה. הקוד הסודי תקף רק להתקשרות זו. כלומר, אין צורך לסמוך על הלקוח שישמור סוד. הקוד הסודי אינו שווה דבר עם סיום ההתקשרות.

כיצד להגן על האתר שלכם בעזרת SSL?

שלב ראשון: הכנת קובץ בקשה לרשיון, על ידי Key Manager

כדי להתקין רשיון דיגיטלי, יש להפעיל את מנהל המפתחות.

תחילה, יש להיכנס ל-IIS, לבחור Properties < Directory Security < Key Manager.
אפשרות נוספת: ללחוץ על הסמל Key Manager בסרגל הכלים.



בחלון זה, סמנו את השירות הרצוי, ובחרו בתפריט Key < Create New Key. בפעולה זו יופיעו מספר חלונות, בהם תתבקשו להקליד פרטים אודות החברה, אותה אתם מבקשים לרשום. כמה מהפרטים דורשים הסבר:

Request File – שם הקובץ, שיווצר על ידי Key Manager, ובו יהיה המידע, שישלח ל-CA.

Key Name – שם המפתח. יכול להיות כשם החברה שלכם, או כל שם אחר.

Password – סיסמת הגנה על המפתח הפרטי. נא לא להקליד 1234!!!

Bits – אורך המפתח המבוקש ב-Bits. הקלידו 512.

Common Name – שם האתר שלכם, כפי שהגולשים יקלידו אותו. לדוגמה: www.sifroni.co.il.

יתר הפרטים פשוטים: שם, כתובת, טלפונים וכדומה.

לאחר שסיימתם להקליד את כל הפרטים, בחרו בתפריט Computers > Commit Changes Now, ותקבלו קובץ בשם שביקשתם. Key Manager, יצר זוג מפתחות (פרטי וציבורי), וכעת יש לשלוח אותם ל-CA כלשהו, כדי להפוך את הבקשה לרשיון בר תוקף.

להלן דוגמה לקובץ המתקבל:

Webmaster: roni@sifroni.co.il
Phone: 03-8484848
Server: Microsoft Key Manager for IIS Version 4.0

Common-name: sifroni.co.il
Organization Unit: Sifroni
Organization: Sifroni E-Commerce Ltd.
Locality: Haifa
State: Israel
Country: IL

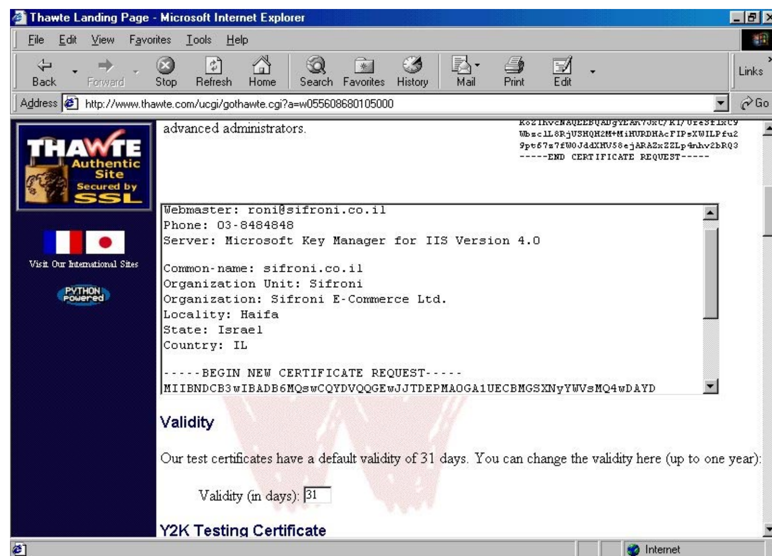
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBNDCB3wIBADB6MQswCQYDVQQGEwJJTDEPMA0GA1UECBMGSXNyYWV
VQQHEwVIYWlmYTEgMB4GA1UEChMXU2lmcm9uaSBFLUNvbW1lcmNlIEx0ZC4
BgNVBAsTB1NpZnJvbmkxFjAUBgNVBAMTDXNpZnJvbmkuY28uaWwwXDANBgk
9w0BAQEFAANLADBIAkEAvJ/aIXgL6E4ECcS+LL5GoCwTHkFqnK7gFzii4U1cLHK
oFS9GcFJP8E2eyS/rNzD4KUxGlE7FWqu3CYt/+bnCQIDAQABoAAwDQYJKoZIhvcN
AQEEBQADQQBNSBMRsOa6r5oL6NV2P0908+k8A8yjs8l90efQ8jq03Wn1DNP7/+ik
+ChoxRh09Kebgp23mjFb+ZBPAQvrPR2I
-----END NEW CERTIFICATE REQUEST-----

את הקובץ שהתקבל נעביר, בדיסקט או דרך הרשת, למחשב הפיתוח.

שלב שני: קבלת רשיון דיגיטלי

יצרנו קובץ בקשה, וכעת עלינו לשלוח את הבקשה לאחת הרשויות המוסמכות. היכנסו לאתר של אחת מהחברות הרשומות למעלה (או אחרות), שלחו אליה את קובץ הבקשה, ורכשו רשיון דיגיטלי. לצרכי לימוד, ניתן בחלק מהחברות לקבל רשיון הדגמה (דמו), ללא תשלום, לתקופה של מספר שבועות. בדוגמה שלפנינו, נקבל רשיון הדגמה מחברת Thawte. ניכנס לאתר החברה: www.thawte.com, נחפש את האפשרות לקבל רשיון הדגמה, ונגיע לדף מילוי הפרטים. הקפידו בשדה Server DNS Name, להקליד את שם האתר שלכם בדיוק, כפי שיקלידו אותו המשתמשים. לדוגמה: www.sifroni.co.il. יתר הפרטים חשובים בעיקר, אם אתם מזמינים רשיון אמיתי. במקרה כזה, הפרטים אכן ייבדקו.

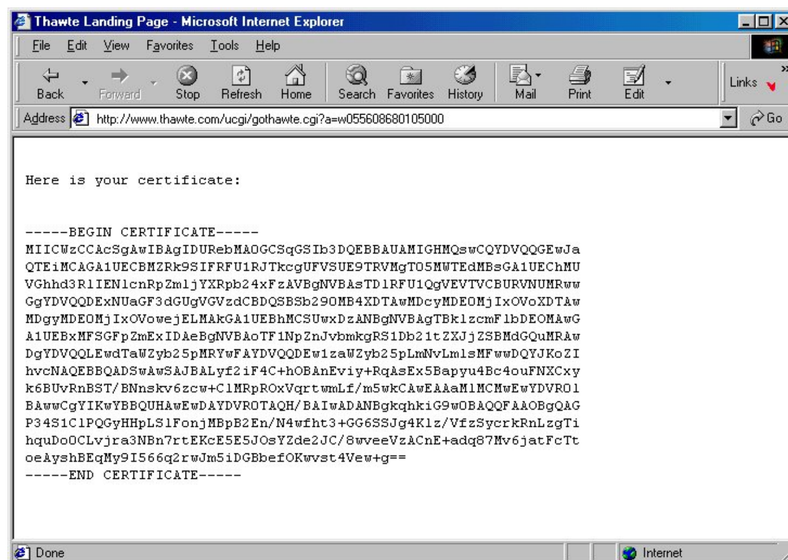
שימו לב! אחד הפרטים שתידרשו לתת הוא סיסמה. רשמו לפניכם את הסיסמה שהקלדתם, כי רק בעזרתה תוכלו להתקין את הרשיון שתקבלו!



דף הזנת הבקשה לרשיון באתר Thawte.com.

נפתח את קובץ הבקשה בעזרת NotePad, או כל עורך טקסטים אחר, ונעביר אותו ללוח. סמנו את כל הקובץ (Select All), ובתפריט העריכה בחרו Copy. נעבור לדפדפן, שכבר פתוח על הדף הנכון באתר, נציב את הסמן בתיבת הטקסט, ונלחץ Ctrl + V, או נבחר בתפריט העריכה Paste.

נאשר את הפרטים, ונקבל דף ובו נתוני רשיון ההדגמה.





הערה:

במקרה של רשיון אמיתי, קבלת הרשיון עשויה לקחת מספר ימים, עקב אימות הנתונים שמתבצע!

נסמן את כל הקטע בדפדפן, מהמילים ---BEGIN CERTIFICATE--- (כולל!!), ועד למילים ---END CERTIFICATE--- ונעתיק אותו ללוח. נפתח מסמך טקסט חדש ב- NotePad, ונדביק לתוכו את תכולת הלוח, כלומר את כל הקטע, שהגיע מהדפדפן. ניתן שם למסמך ונשמור אותו בדיסק. כך:

thwtekey.txt - Notepad

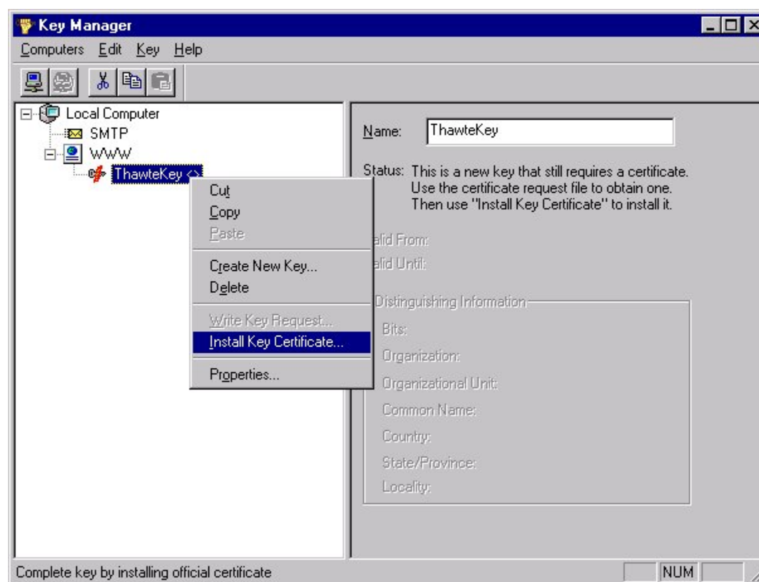
File Edit Search Help


```
-----BEGIN CERTIFICATE-----
MIICWzCCAcSgAwIBAgIDURebMA0GCSqGSIb3DQEBBAUAMIGHMQswCQYDVQQGEwJa
QTEiMCAGA1UECBMZRk9SIFRFU1RJTkcgUFVSUE9TRVMgT05MWTEdMBsGA1UEChMU
VGhhd3RlIENlcnRpZmljYXRpb24xFzAVBgNVBAsTDlRFU1QgVEVTVCBURVNUMRww
GgYDVQQDExNUaGF3dGUgVGVzdCBDQSBSb290MB4XDTAwMDcyMDE0MjIxOVoXDTAw
MDgyMDE0MjIxOVowejELMAkGA1UEBhMCSUwxDzANBgNVBAgTBklzcmFlbDEOMAwG
A1UEBxMFSGFpZmExIDAeBgNVBAoTF1NpZnJvbmkgRS1Db21tZXJjZSBMdGQuMRAw
DgYDVQQLEwdTaWZyb25pMRYwFAYDVQQDEw1zaWZyb25pLmNvLmlsMFwwDQYJKoZI
hvcNAQEBBQADSwAwSAJBALyf2iF4C+hOBAnEviy+RqAsEx5Bapyu4Bc4ouFNXCxy
k6BUvRnBST/BNnskv6zcw+ClMRpROxVqrtwmLf/m5wkCAwEAAaMlMCMwEwYDVR0l
BAwwCgYIKwYBBQUHAwEwDAYDVR0TAQH/BAIwADANBgkqhkiG9w0BAQQFAAOBgQAG
P34S1ClPQGyHHpLSlFonjMBpB2En/N4wFht3+GG6SSJg4Klz/VfzSycrkRnLzgTi
hquDo0CLvjra3NBn7rtEKcE5E5JOsYZde2JC/8wveeVzACnE+adq87Mv6jatFcTt
oeAyshBEqMy9I566q2rwJm5iDGBbefOKwvst4Vew+g==
-----END CERTIFICATE-----
```


שלב שלישי: התקנת רשיון דיגיטלי

כשקובץ זה בידנו, נעבור לשרת. נעביר את קובץ הרשיון לתיקיה כלשהי, ונפתח את Key Manager.

המפתח שיצרנו מופיע, אך יש עליו פס אדום. כלומר: זוהי רק בקשה למפתח, ולא מפתח אמיתי. כעת נתקין על הבקשה את המפתח שקיבלנו. לחצו לחיצה ימנית על סמל המפתח, ובחרו Install Key Certificate.



ייפתח חלון חיפוש, ותתבקש למצוא את קובץ הרשיון. מצא אותו ואשרו. יופיע חלון סיסמה. זו אותה סיסמה, שבחרתם בעת יצירת הבקשה. הקלידו אותה. כעת תתבצע בדיקת השוואה בין הנתונים שביקשתם, לבין אלה החבויים בקובץ. כלומר: קובץ יכול להיות מותקן רק על הבקשה, שיצרה אותו!

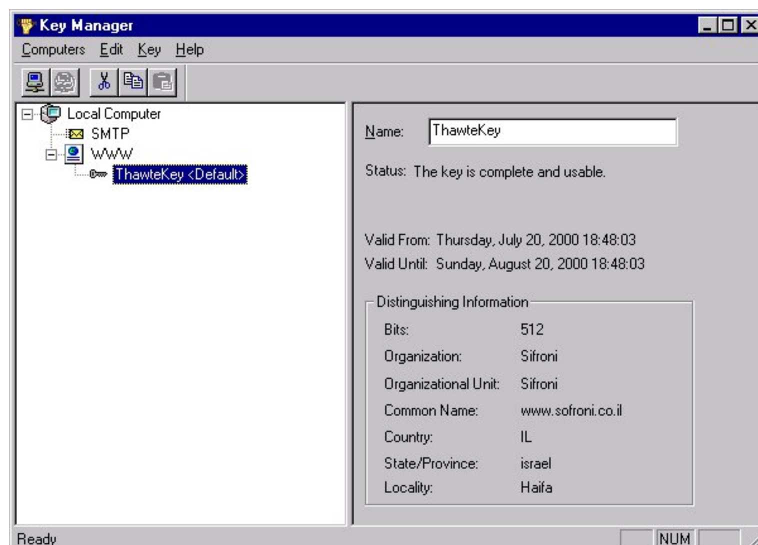
כעת יופיע חלון, בו תוכלו להצמיד את הרשיון לכתובת IP מסוימת. הצמידו את הרשיון לכתובת IP של השרת שלכם.



הערה חשובה:

כתובות IP רבות, יכולות להיות מוצמדות לרשיון אחד, אך לא יותר מרשיון אחד יכול להיות מוצמד לכתובת!

אם הכל עובר בשלום, יותקן המפתח, ובחלון הפרטים בצד ימין, תוכלו לראות את פרטי בעל הרשיון:



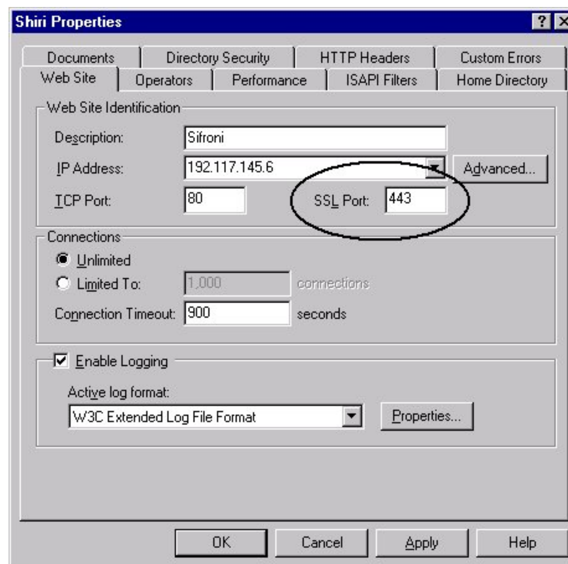
לסיום, יש לבחור ב- Commit Changes Now מתפריט Computers, כדי לשמור את השינויים.

שלב רביעי: הגנה על קבצים

יפה, התקנתם מפתח SSL בשרת! כעת נותרה רק משימה קלה אחת: להשתמש בו.

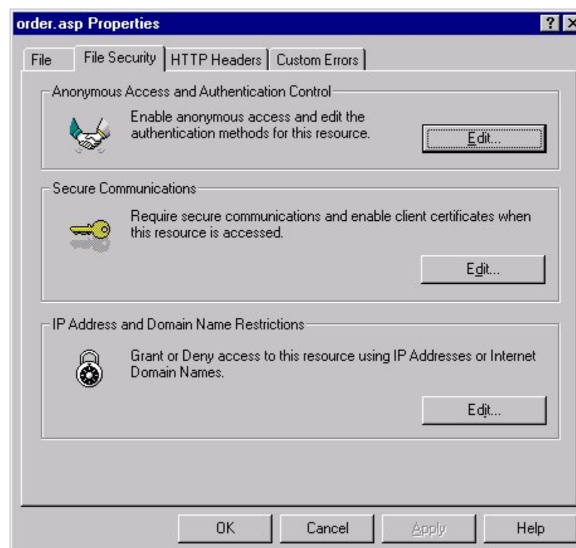
יש להחליט על אילו דפים ברצונכם להגן. תאמרו בוודאי: מה הבעיה? נגן על כולם! אלא שהדברים אינם כה פשוטים. עבודה ב-SSL מטילה על השרת עומס חישובי עצום, בהשוואה לדף רגיל. לכן נכניס למטריית ההגנה רק את הדפים עליהם חובה להגן. כמו במקרה שלנו, דף ביצוע ההזמנה. על דפי חיפוש הספרים שלנו אין צורך להגן. כל מי שרוצה מוזמן לעיין בספרים.

שלב ראשון הוא לאפשר את Port 443 (זהו ה-Port המשוייך לפרוטוקול SSL). היכנסו ב-IIS למאפייני השרת, וקבעו את מספר ה-Port לו צריך להאזין (כאמור, 443). בדקו עם רשות הרישוי שלכם את מספר ה-Port.

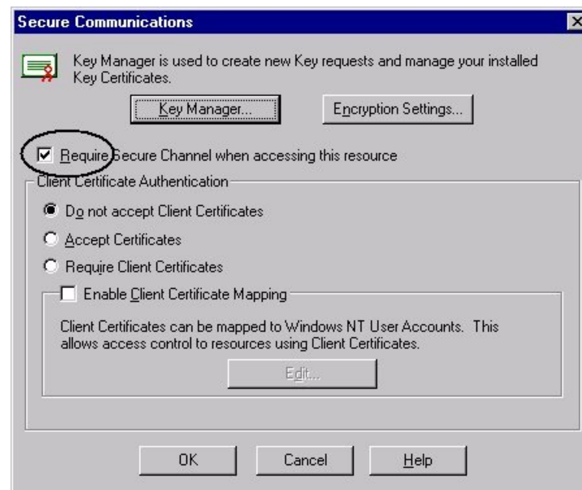


נחליט על אילו דפים אנו רוצים להגן, ועבור כל אחד מהם, נבצע את התהליך הבא:

נפתח את מנהל ה-IIS, ונבחר קובץ עליו אנו רוצים להגן. בדוגמה שלנו, בחרתי להגן על דף הזמנה. זו בחירה סבירה, מפני שבדף זה, מקליד הלקוח את פרטי כרטיס האשראי שלו. ניתן כמובן להגן על יותר מדף אחד. נמצא את הקובץ, נלחץ עליו לחיצה ימנית ונבחר Properties. בחלון זה נבחר File Security. הנה כך:



נלחץ על Edit בתיקיה Secure Communications, והחלון הבא יופיע:



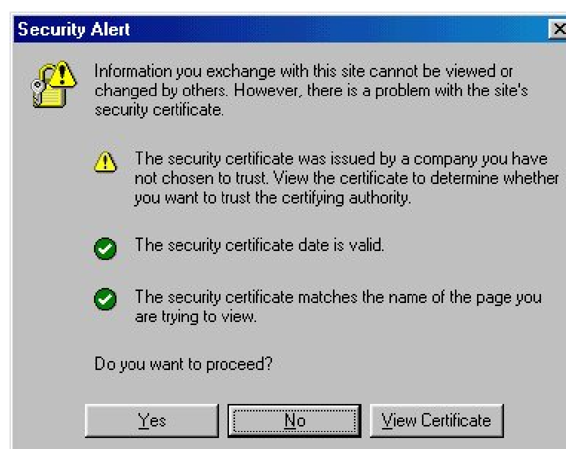
נסמן את תיבת הסימון ...Require Secure Channel ונאשר. זה הכל.

דבר נוסף שיש לזכור: בגלל החלפת הפרוטוקולים, אי אפשר לקרוא לדף מוגן SSL מתוך דף שאינו מוגן, בקריאה יחסית, כמו <A HREF=order.asp>, אלא צריכים לציין את המסלול המלא:

<A HREF=https://www.sifroni.co.il/order.asp>Order Now!</A>

כאשר קוראים לדף מוגן מתוך דף מוגן אחר, אין החלפת פרוטוקולים, ואז מותרת הפניה יחסית.

בדף showbasket.asp, הקורא לדף ההזמנה order.asp, נשנה את הקישור, כך שיראה את המסלול המלא והאמיתי של הדף. ניכנס לאתר, נזמין ספרים, ונלחץ על הלחצן "לקופה". נקבל הודעה:



זוהי הודעה סטנדרטית לפני כניסה לדף מוגן SSL. לפי הודעה זו, הלקוח יכול לדעת אם אכן האתר מאובטח, ובאיזו רמה. הוא יכול גם לקבל את פרטי הרשיון, ולראות מי הוציא אותו!

נאשר את ההודעה, ונקבל את דף ההזמנה:

https://www.shiri.co.il/bookstore/order.asp - Microsoft Internet Explorer

Address https://www.shiri.co.il/bookstore/order.asp

חיפוש קניות חברים מבצעים אודות

שם מלא	שלמה גולדשטיין
Email	Shlomo@zahav.net.i
כתובת	הרצל 17 תל אביב 2323
מסי זהות	123456789
מסי טלפון	052-998877
כרטיס אשראי	American Express
מספר כרטיס	26783746372618
תוקף	חודש 3 שנה 2002
לתשלום	170.00 ש"ח

המשך!

Internet

שימו לב לשני השינויים שחלו בדף:

1. הכתובת מתחילה ב- https://.
2. בתחתית הדפדפן מופיע מנעול, המסמל עבודה במצב מאובטח.



שאלה: ומה אם לא הבנתי כלום מהפרק הזה, ובכל זאת אני רוצה להגן על השרת שלי?

תשובה: יש חברות, שתמורת סכום סמלי, תשמחנה לאבטח את השרת שלך. בדרך כלל, הרעיון הוא, שהדפים המוגנים יישבו בשרת של החברה המארחת, וכך אתה נהנה מ"מטריית" האבטחה שלה.

חברה כזו לדוגמה: http://www.sslservices.com.

פרק 25

עברית באינטרנט

עברית היא שפה קשה, ועוד יותר מכך קשה באינטרנט. שתי בעיות עיקריות מפריעות לכתיבת עברית בדפי אינטרנט. הראשונה: צורת האותיות, הגופנים. הבעיה השנייה: כיוון הכתיבה, מימין לשמאל. קיימות כמה שיטות להתמודדות עם נושא העברית. בפרק זה נלמד את שיטות הפתרון הקיימות, וכמובן נשים דגש על צורת היישום של פתרונות אלה בעבודה עם Visual InterDev.

שיטת התמונה

בשיטה זו, כותבים את הטקסט בתוכנה גרפית כלשהי, ושומרים אותו כקובץ תמונה רגיל, כמו JPG, GIF או אחרים. את התמונה משלבים בדף ככל תמונה אחרת:

```
<IMG SRC="MyText.JPG">
```

יתרונות

- הטקסט ייראה בדיוק כפי שכתבנו אותו. לא תהיה שום אי ודאות לגבי צורת האותיות או שבירת השורות.
- אין צורך להתקין גופנים.
- אין צורך בתמיכה עברית בדפדפן. גם בשבדיה יוכלו לקרוא את הדף שלנו, אם זה מעניין אותם.

חסרונות

- זמן ההורדה של הטקסט יהיה ארוך, מכיון שזו תמונה. "משקלה" של תמונה גדול, בדרך כלל, פי כמה ממשקלו של הטקסט הכתוב בה.
- אי אפשר יהיה להעתיק (Copy) את הטקסט מהאתר, ולהעבירו ליישום אחר.
- מבחינתנו, כמתכנתי ASP, השיטה פסולה לחלוטין, מפני שהמידע שלנו אינו סטטי, אלא מגיע ממסד נתונים. לכן, יצירת תמונה מכל טקסט אינה מעשית.

עברית לוגית

שיטה זו נקראת לעיתים גם השיטה המשתמעת - Implicit. בשיטה זו, נכתבות האותיות בסדר ה"נכון" שלהן. כלומר, בסדר בו הן מקבלות משמעות. למשל, המשפט "שמי dan", ייכתב כך: ש, אחר כך: מ, אחר כך: י, אחר כך: רווח, אחר כך: d, אחר כך: a, אחר כך: n. מערכת ההפעלה המציגה את הטקסט, צריכה לדעת באות d, שעברנו לאנגלית, וכעת צריך להפוך כיוון. שיטה זו, אומצה על ידי Microsoft, והיא נתמכת באופן מושלם ב- Internet Explorer 5.0.

יתרונות

- זו שיטה קלה מאוד לעבודה. אם תבחרו בשיטה הלוגית, תוכלו להקליד טקסטים עבריים ישירות לתוך חלונות העריכה של- Visual InterDev.
- ניתן לערוך בקלות את הטקסט, להעתיק (Copy, Paste) קטעים מהאתר ואליו, ולהדפיס ללא בעיות. אם קיבלתם מסמך Word מלקוח, ועליכם לשלב אותו בדף HTML, זו משימה של מספר דקות.

חסרונות

- יש רק חיסרון אחד: השיטה הלוגית נתמכת רק על ידי Internet Explorer של מיקרוסופט. זה אמנם, רוב השוק, נכון לרגע כתיבת שורות אלה, אבל בהחלט לא כולו.

עברית ויזואלית

בשיטה הוויזואלית (החזותית), נכתבות אותיות הטקסט באותו סדר שבו הן צריכות להיות מוצגות על מסך המחשב. למשל, המשפט "שמי dan" ייכתב כך: d, אחר כך: a, אחר כך: n, אחר כך: רווח, אחר כך: י', אחר כך: מ', אחר כך: ש'. ממש באותו סדר, שהיינו עושים זאת במכונת כתיבה, הפועלת משמאל לימין.

חסרונות

- שיטה זו מצריכה כתיבה הפוכה בעת עריכת דף HTML או ASP.
- אי אפשר לכתוב עברית שוטפת לתוך חלונות העריכה של Visual InterDev, כמו שעשינו במהלך הדוגמאות של ספר זה.
- אי אפשר להעתיק טקסט מתוך אתר, שנבנה בעברית ויזואלית ולהדביקו ליישום אחר. הוא יועתק הפוך.
- יש בעיות קשות בהדפסת דפים מאתר כזה.

יתרונות

- שני הדפדפנים הנפוצים ביותר, Netscape ו- Explorer, תומכים בשיטה זו. לכן, זו בדרך כלל, השיטה המועדפת.
- אין צורך במערכת הפעלה עברית. גם משתמשים בחו״ל, יכולים לראות נכון אתרים בעברית, על ידי התקנת גופן בלבד.

בדפדפנים של חברת Netscape אין תמיכה בעברית. כדי לאפשר עבודה בעברית, נכתבו גופנים מיוחדים למטרה זו בלבד. לדוגמה: AD Hebrew ,Globes Web ואחרים. גופנים אלה יעזרו לדפדפן Netscape, רק אם הדף נכתב בעברית ויזואלית.

כדי לגרום לדפדפן לזהות מייד, שמדובר בעברית ויזואלית, יש להוסיף את השורה הבאה בתוך כותרת הדף:

```
<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=iso-8859-8">
```

מהסיבה האמורה, של תאימות לשני הדפדפנים המובילים, ותוך השלמה עם המגרעות, התקבלה שיטת העברית הוויזואלית, כתקן דה-פאקטו לאתרים בעברית.

יש להודות, שהשיטה של מיקרוסופט טובה יותר, נכונה יותר ועושה חיים קלים יותר לבוני האתרים בכל הרמות, גם הדפדפן שלהם מוצלח יותר, וזו החברה היחידה שדאגה להתאים את מוצריה לעברית! לכן, אם אתם כותבים יישום אינטראנט, בו יש לכם שליטה על הדפדפן בו משתמשים העובדים, כדאי לעבוד בשיטה זו. אם לא, וכל עוד לא סיימה מיקרוסופט לחלוטין את השתלטותה על העולם, כדאי להתחשב גם במשתמשי Netscape, ולהשתמש בעברית ויזואלית.

שיטות להפיכת העברית

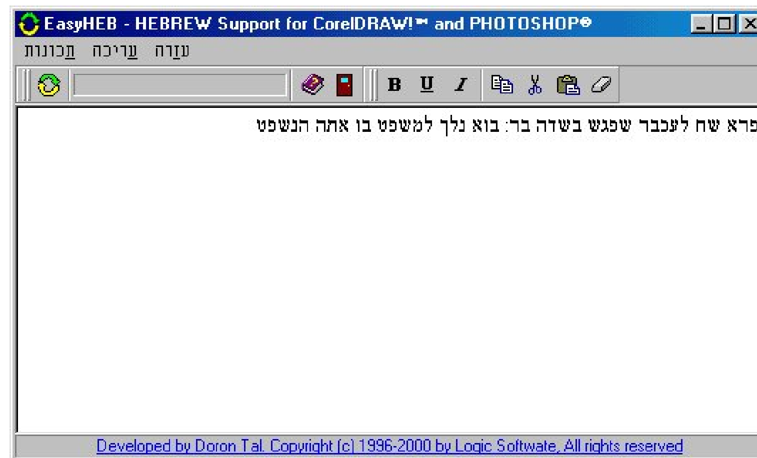
החלטנו לעבוד בעברית ויזואלית, וזה אומר שיש להפוך כל טקסט לפני כתיבתו בדף. ישנם כלים רבים לכך, חלקם ניתנים ללא תשלום, וחלקם בתשלום. להלן סקירה קצרה של העיקריים שבהם.

קיוטקסט

מעבד תמלילים נפלא. לדעת רבים – הטוב ביותר שקיים בעברית. בין יתר תכונותיו המופלאות, יודע קיוטקסט לייצר דפי HTML. כולל קישורים, רקעים, תמונות וכל השאר. בנושא זה, לקיוטקסט יש שני יתרונות גדולים על פני Word: הראשון - קובץ ה-HTML, שנוצר הוא פשוט יחסית, וניתן לטיפול ידני. ב-Word, נוצר קובץ שקשה מאוד לתחזוקה. היתרון השני חשוב יותר - קיוטקסט יודע לייצא קובץ בעברית לוגית או ויזואלית, לפי בחירתכם! Word, כמובן תומך רק בשיטת מיקרוסופט, כלומר עברית לוגית. קיוטקסט הוא מוצר מסחרי, וניתן לקבל עליו פרטים באתר החברה www.qtext.com.

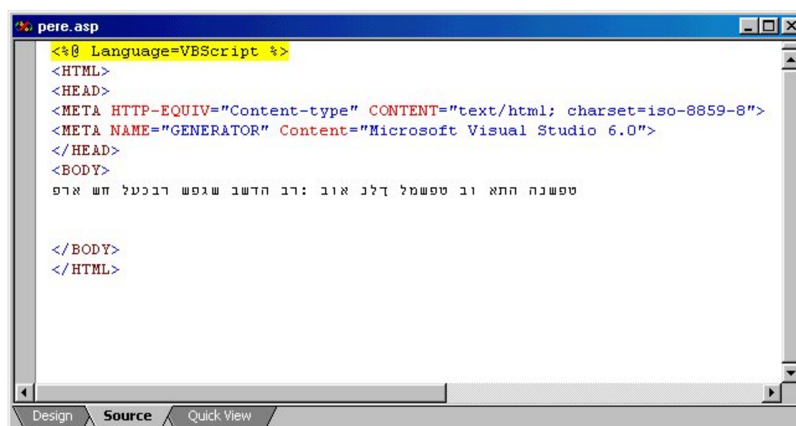
Easy Heb

זוהי תוכנה קטנה ונחמדה, הניתנת ללא תשלום, וכל תפקידה הוא להפוך קטעי טקסט. ניתן להוריד אותה מאתר האינטרנט http://logisoft.netvision.net.il. לאחר התקנת התוכנה, צריך להקליד לתוכנה את הטקסט (או להעתיק אותו ממסמך Word), וללחוץ על לחצן ההפיכה:



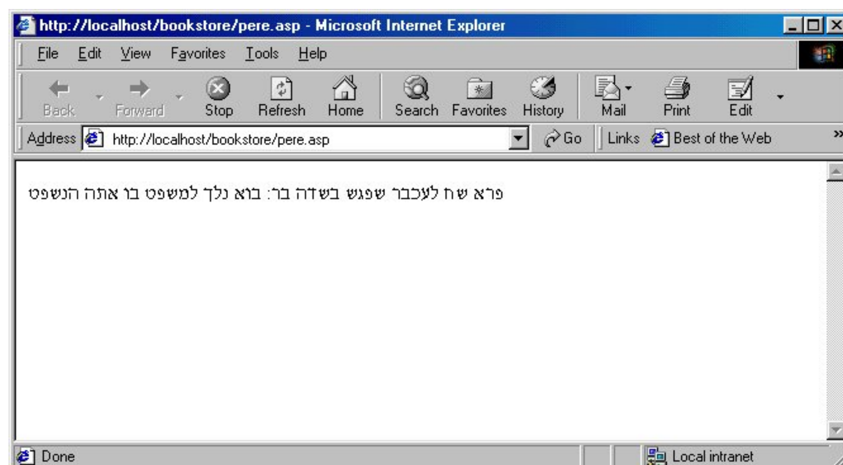
לחיצה על לחצן ההפיכה מכניסה את כל הטקסט, כשהוא הפוך ומוכן, לתוך הלוח (ClipBoard). התוכנה יורדת לשורת המשימות, ומחכה לשימוש נוסף. כעת עוברים לדף HTML או ASP שלנו, ובתפריט העריכה בוחרים Paste. ניתן גם להקיש Ctrl+V.

כך נראה חלון העריכה של Visual InterDev, לאחר הדבקת הטקסט:



הטקסט הפוך, קשה לקריאה וקשה מאוד לעריכה. במקרה של שורות רבות, כדאי לכתוב
 בסוף כל שורה.

כאשר נריץ את הדף, יזהה הדפדפן שמדובר בעברית ויזואלית (בזכות שורת META), ויציג את הטקסט נכון:



כלים נוספים להפיכת שורות לעברית ויזואלית:

- עברית 97 של בשן. ניתן להורדה מהכתובת: www.hebrew97.bashan.co.il.
- זיקית. ניתן להורדה מהכתובת: www.zikit.co.il.
- תוסף ל-Word, שנקרא E2. ניתן להורדה מהכתובת: www.evrit.co.il.

שבירת שורות

נקודה חשובה שיש להתחשב בה, היא שבירת שורות. כזכור, בשיטה הוויזואלית נכתבות האותיות לפי סדר הצגתן, משמאל לימין. לכן כאשר נגמרת השורה, עוברות לשורה הבאה המילים הימניות ביותר בשורת המקור, שהן המילים הראשונות. זה כמובן יוצר תוצאה, שאינה קריאה. לדוגמה, ננסה לכתוב את המשפט הבא:

אני מסוגל לבאר את כל השירים שהומצאו אי פעם, וגם רבים מאלה שעדיין לא הומצאו.

נהפוך אותו בעזרת Easy Heb, ונקבל:

ואצמוה אל ןיידעש הלאמ םיבר םגו ,םעפ יא ואצמוהש םירישה לכ תא ראבל לגוסמ ינא.

אם בדף יהיה מקום רק לחלק מהשורה, תיראה התוצאה כך:

השירים שהומצאו אי פעם, וגם רבים מאלה שעדיין לא הומצאו.

אני מסוגל לבאר את כל

תוצאה זו שגויה לחלוטין, כמובן, למרות שהמילים נראות בסדר. כדי להימנע מכך, עלינו לשבור את השורות בעצמנו, תוך התחשבות באורך השורה העומד לרשותנו. אם נכתוב בקוד המקור כך:

**סעפ יא ואצמוהש סירישה לכ תא ראבל לגוסמ ינא
**

**ואצמוה אל ןיידעש הלאמ סיבר סגו
**

התוצאה תהיה טובה יותר:

אני מסוגל לבאר את כל השירים שהומצאו אי פעם

וגם רבים מאלה שעדיין לא הומצאו

מכאן שיש למצוא את אורך השורה העומדת לרשותנו, ולשבור את השורות באורך הקטן מזה במעט. כדאי להביא בחשבון גופנים מעט שונים במחשב הלקוח. בגופנים אלה אולי יהיו האותיות רחבות מעט יותר. לכן כדאי לקחת מקדם בטחון נוסף, של בערך 10%.

הפיכה דינמית של נתונים מטבלאות

כל עוד היה הטקסט קבוע ויציב, היה מצבנו טוב יחסית. הפעלנו את Easy Heb, או כלי אחר, הפכנו, ושתלנו את התוצאה בדף שלנו. בדקנו את התוצאה, ואם היה צורך, ביצענו תיקון. המצב מסתבך מעט, כאשר מדובר בנתונים, שמגיעים מתוך טבלה. היזכרו בדף BookDetail.asp. נתוני הספר נלקחו ממסד הנתונים, ונשפכו לתוך הדף. יש צורך בתוכנית, שתבצע הפיכה בזמן אמת של טקסט רגיל, לטקסט בעברית ויזואלית. תוכנית כזו יכולה להיות פונקציה הכתובה ב- VBScript, ונמצאת בדף ASP רגיל, שאנו מכילים בעזרת הפקודה Include בכל אחד מהדפים שלנו. אפשרות אחרת, טובה ומקצועית יותר, היא להכין רכיב COM, שיבצע את העבודה. רכיבי COM מהירים הרבה יותר מדפי ASP, וכדאי להעביר אליהם משימות מוגדרות עתירות חישובים, בדיוק כמו במקרה זה.

האם השרת שלי תומך בעברית?

שרתי אינטרנט אינם תומכים בשפות, הם תומכים בסיביות (Bits). אין להם מושג מה מכילים דפי האינטרנט המאוחסנים בהם, ומסדי הנתונים המקושרים אליהם. זו בשורה טובה, שמשמעותה: כן, השרת שלך תומך בעברית. רק כתוב את הנתונים נכון, וקרא אותם נכון.

טיפים לעבודה עם עברית:

1. בראש כל דף כתוב כך (בשורה הראשונה!!!):

```
<%@ Language=VBScript Codepage=1255 %>
```

2. החלט באילו משיטות העברית ברצונך לעבוד, וגרום לדפדפן לעבוד בצורה שהחלטת.

מקומה של שורת META הוא בין התג <head> לתג </head>.

לעברית לוגית:

```
<META HTTP-EQUIV="Content-Type" content="text/html; charset=windows-1255">
```

לעברית ויזואלית:

```
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-8">
```

3. אם אתה עובד ב- SQL Server, העדף את סוגי הנתונים ששמותיהם מתחילים ב-n.

nchar במקום char, nvarchar במקום varchar, וכן הלאה. המשמעות היא שימוש ב- Unicode.

Unicode היא שיטת ייצוג אותיות בה שומרים כל תו בשני בתים במקום אחד ב-ASCII (כן, זה אומר תצרוכת דיסק כפולה לאותם שדות!). הבית הראשון שומר את קוד האות כרגיל, והשני את קוד ה"דף". כלומר: לפי איזו מפה לפרש את קוד האות? לכל מדינה או אזור יש מפה, CodePage משלו. הדף העברי, לדוגמה, מספרו 1255.

4. תוכנת Access לדורותיה תומכת בעברית ללא קושי. אם אינך מצליח לראות עברית בדף ASP, והנתונים הם בעברית, מן הסתם התקלה היא בדף ה-ASP.

זהו. השרת שלך תומך בעברית!

פונקציות המרה

יש פונקציות המרה רבות, חלקן ניתן להורדה חינם מהאינטרנט. במקרה הכינותי מראש פונקציה פשוטה יחסית, שכתב אלעד רוזנברג מ-www.myscript.co.il. זהו מקום טוב להתחיל בו את חיפושיכם בנושא זה.

הרעיון של הפונקציה די פשוט: סורקים את הטקסט אות אות, ובונים מחרוזת חדשה. עבור כל אות בודקים: אם זו אות עברית, משרשרים אותה מצד שמאל לטקסט שכבר קיים. אם זו אות אנגלית, משרשרים לצד ימין.

```
<%
option explicit

Function reverseHebrew(theStr)
dim theStrlength  'string length
dim position  'counter
dim OneLetter  'one letter from the string
dim myString  'the reverse string
```

```
'אורך המחרוזת
	theStrLength = len(theStr)
	position = 0
	'ריצה על המחרוזת מהסוף להתחלה
	For Position = theStrLength To 1 Step -1
		'בכל ריצה על המחרוזת אנו מזהים תו אחד
		'עד שמגיעים לסוף המחרוזת
		OneLetter = Mid(theStr, Position, 1)
		'בדיקה לגבי כל אות האם היא אנגלית
		'כל תו מסומן על ידי מספר אסקי
		'אלו המספרים של האותיות הלטיניות - קטנות וגדולות
		Select Case CInt(Asc(OneLetter))
			case 65,66,67,68,69,70,71,72,73,74,75,76,77,78,_
			   79,80,81,82,83,84,85,86,87,89,90,97,98,99,100,_
			   101,102,103,104,105,106,107,108,109,110,_
			   111,112,113,114,115,116,117,118,119,120,121,122
			'אם מדובר באנגלית תחזיר את המחרוזת בסדר הרגיל ולא ההפוך
			myString = OneLetter & MyString
			case else
			'במקרה ומדובר בעברית תשרשר את המחרוזת בהיפוך
			MyString = MyString & OneLetter
		end select
	next
	'(הפונקציה מחזירה מחרוזת הפוכה (או לא אם מדובר באנגלית
	reverseHebrew = myString
end function

%>
```

את הפונקציה הזו תמצאו בתקליטור המצורף בקובץ בשם reverseHebrew.asp.

כעת נכתוב דף פשוט לבדיקת הפונקציה. הוסיפו דף בשם TestHebrew.asp. ראשית, זהו דף בעברית ויזואלית, ולכן נוסיף לו שורת Meta :

```
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=iso-8859-8">
```

כדי שנוכל להשתמש בפונקציה, נעשה include לקובץ.

```
<!-- #include file = "reverseHebrew.asp" -->
```

לאחר מכן, בתוך תגית Body, נכתוב שורה בעברית, ונמיר אותה בעזרת הפונקציה.

```
response.write reverseHebrew("שמי פיליפ, ובאנגלית זה philip")
```

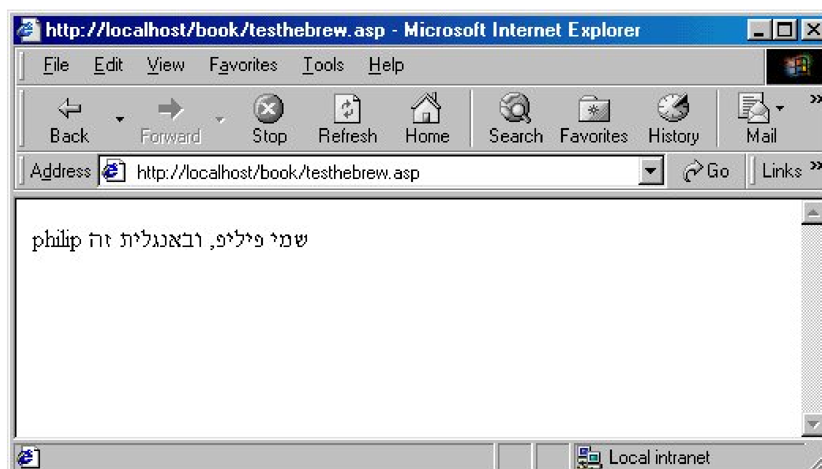
הכוונה היא לכתוב שורה ובה עברית ואנגלית מעורבבים יחד, כדי לבדוק את הפונקציה. היא כמובן תעבוד היטב גם עם עברית בלבד.

הנה הקוד של כל הדף יחד :

```
<%@ Language=VBScript %>
<!-- #include file = "reverseHebrew.asp" -->
<HTML>
<HEAD>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=iso-8859-8">
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>
<%
response.write reverseHebrew("שמי פיליפ, ובאנגלית זה philip")

%>

</BODY>
</HTML>
```



לבדיקה, לחצו לחיצה ימנית על הדפדפן, בחרו Encoding, וודאו שהדפדפן אכן עבר לעברית ויזואלית.

פרק 26

איתור תקלות

פעמים רבות בעת עבודתכם בפיתוח דפי ASP, תיתקלו בהודעות שגיאה. קראו אותן. לעיתים קרובות יש בהודעה רמז, אם לא אמירה מפורשת, בדבר סיבת התקלה. למרות זאת, ריכזתי עבורכם את הודעות השגיאה הנפוצות ביותר, עם הסבר לכל אחת, ועם הצעות לפתרון.

.1

```
Response object error 'ASP 0156 : 80004005
Header Error
Loclahost/project1/pagetest.asp,  line 48
The HTTP headers are already written to the client browser. Any HTTP header
modifications must be made before writing page content.
```

הסבר

בפרוטוקול HTTP, שבו אנו עובדים, יש לכל תשדורת כותרת - Header. כאשר אתם שולחים תג HTML כלשהו, או אפילו טקסט רגיל, נשלחת לפניו, אוטומטית, כותרת.

פקודות ASP מסוימות, משתמשות באותה כותרת, ויכולות להתבצע רק תוך כדי שליחת כותרת. לכן הן לא תוכלנה להתבצע אם כותרת כבר נשלחה.

הפקודות הן:

1. כל הפקודות שקשורות ל- Response.Cookies.
2. Response.Redirect.

פתרון

הבנת מקור הבעיה, במקרה זה, מקרבת אותנו מאוד לפתרון. פשוט, בצעו פקודות רגישות אלו ראשונות. יש לכם עוגיה לשלוח? שלחו אותה מייד בתחילת הדף, ורק אחר כך, ציירו טבלאות, כתבו כותרות וכל היתר.

שימו לב! כאשר אתם יוצרים דף ASP חדש, נוצר אוטומטית "שלד" HTML. ואתם בדרך כלל, כותבים בתוך מסגרת ה- BODY. שלד זה לבדו גורם לשליחת כותרות.

לעיתים, אינכם צריכים את השלד כלל. לדוגמה, בדף addtobasket.asp, אין צורך ב-HTML כלל. לכן יש למחוק את כל השלד ולכתוב רק את מה שצריך.

אם בכל זאת יש צורך בקוד HTML, בצעו את הפקודות הרגישות **ראשונות**, לפני ה-HTML.

.2

```
Microsoft OLE DB Provider for ODBC Drivers error '80004005'
[Microsoft][ODBC Microsoft Access Driver] Table 'visitors' is exclusively locked by user
'Admin' on machine 'RONEN'.
/bookstore/orderproc.asp, line 58
```

הסבר

הטבלה פתוחה במצב עיצוב, ולכן נעולה לכל המשתמשים.

פתרון

עברו לאקסס, וסגרו את הטבלה.

.3

```
Microsoft OLE DB Provider for ODBC Drivers error '80004005'
[Microsoft][ODBC Microsoft Access Driver] Field 'visitors.address' cannot be a
zero-length string.
/bookstore/orderproc.asp, line 58
```

הסבר

ניסיתם להכניס לשדה בטבלה ערך ריק, ושדה זה מוגדר כרגע ככזה, שאסור להכניס לו ערכים ריקים.

פתרון

תחילה, אם הגעתם להודעה זו, סימן שלא ביצעתם בדיקת שגיאות כמו שצריך. הייתם צריכים "לתפוס" את העובדה, שהשדה ריק ולא לחכות, שמסד הנתונים "יצעק".

אפשרות אחת היא כאמור, ללכוד את השגיאה ולהודיע בנימוס: "שכחת להקליד שם, אנא הקלד..."

אפשרות שנייה, היא להחליט שזהו שדה אופציונלי. לא רוצה? לא צריך! נעבור ל-Access ונשנה את הגדרת השדה לכזה, שמוכן לקבל גם ערכים ריקים.

סמנו את הטבלה, לחצו על Design:

בחלק העליון סמנו את השדה הרצוי, ואז בחלק התחתון שנו את ההגדרה Allow Zero Length ל-Yes.

שמרו את הטבלה.

4.

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
[Microsoft][ODBC Microsoft Access Driver] Syntax error (missing operator) in query
expression ........
/bookstore/orderproc.asp, line 67
```

הסבר

שגיאה בשאילתת SQL.

פתרון

דרך טובה להתמודדות עם תקלה זו, היא להדפיס את השאילתה לפני הפעלתה. כלומר: מייד עם סיום הבנייה של השאילתה, כתבו:

```
Response.Write sql
```

(בהנחה, ששם המשתנה שלכם הוא sql). כתיבה זו תהיה כמובן, רק לצורך איתור השגיאה. לאחר התיקון – העלימו את השורה.

או. קיי. כתבנו, והנה אנו רואים את השאילתה. היכן השגיאה?

בדרך כלל, תהיה השגיאה בולטת לעין גם אם אינכם אלופי SQL.

1. בדקו שמות שדות. האם השמות בשאילתה מתאימים לשמות בטבלה?
2. זכרו: ערך המוכנס לשדה מחרוזת (String), חייב להיות מוקף גרשיים. לערך מספרי אסור להיות מוקף גרשיים. ככה זה.
3. השוו לשאילתה בספר.
4. צלצלו לחבר טלפוני...

פתרון נוסף

תקלה נפוצה, שגורמת לאותה הודעת שגיאה, היא גרש בתוך המחרוזת המוקלדת. כזכור לכם, אנו משתמשים בגרשיים לפני ואחרי כל שדה מחרוזת. אם המשתמש הקליד משהו כמו דר' סלומון, או מקדונלד'ס ובניו, הגרש מתפרש כסוף המחרוזת, ומה שנשאר גורם לשגיאה. מה עושים? למרבה המזל, SQL "מבין" 2 גרשים כגרש אחד. 2 גרשים לעומת זאת אינם מסמנים סוף מחרוזת. כך נפתרת הבעיה. כל מה שצריך הוא, לפני פעולת insert להחליף כל גרש ב-2 גרשים. עושים זאת כך:

```
fname = Replace(Request.form("fname"), "'", "''")
```

כעת ניתן להכניס את המחרוזת fname לטבלה ללא חשש.

5.

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e10'
[Microsoft][ODBC Microsoft Access Driver] Too few parameters. Expected 1.
/bookstore/personal2.asp, line 35
```

הסבר

יש לכם טעות בשם שדה. בתוך שאילתת SQL נתקל אקסס בשם שדה, שאינו מוכר לו. הוא חושב שזה שם של פרמטר, ומתלונן על שלא סיפקתם את ערך הפרמטר.

פתרון

בדקו את השאילתה שלכם, ודאו שכל שדה שכתבתם אכן מופיע בדיוק כך בטבלה. במיוחד היזהרו ברווחים (אסור בכלל!), קווים תחתונים וכן הלאה. הימנעו משמות שדות המכילים מקף, כמו user-id. מסד הנתונים ינסה לבצע פעולת חיסור בין user לבין id. בדרך כלל, לא יהיו שדות כאלה, ותקבלו הודעת שגיאה על שם שדה. אם קיימים שדות כאלה, תקבלו את תוצאת החיסור. זהו באג שקשה מאוד לזהות את מקורו.

.6

```
Microsoft OLE DB Provider for ODBC Drivers error '80004005'
[Microsoft][ODBC Microsoft Access Driver] Operation must use an updateable query.
/lolueyes/Index.asp, line 13
```

הסבר

ישנן כמה סיבות להודעה זו. הנפוצה ביותר היא הרשאות גישה לא נכונות למשתמש האינטרנט. מדובר בהרשאות גישה ברמת מערכת ההפעלה, והתקלה מתרחשת בעיקר בשרתי NT העובדים ב- NTFS.

פתרון

בדקו את הרשאות הקובץ, ותנו למסד הנתונים הרשאות קריאה וגם כתיבה (R/W).

.7

```
ADODB.Fields error '800a0cc1'
ADO could not find the object in the collection corresponding to the name or ordinal
reference requested by the application
```

הסבר

בדרך כלל, הודעה זו מופיעה עקב טעות בשם שדה בקריאה או כתיבה ל- Recordset.

פתרון

בדקו את הקוד שלכם בשורה המדוברת, ובדקו כל שדה שנמצא בתוך (".....")rs. שם, קרוב לוודאי תמצאו את התשובה.

נספחים ואינדקס

לפניך שני נספחים ואינדקס לספר כולו, כולם באנגלית. שים לב, הנספח הראשון מתחיל בעמוד 3 שנמצא בסוף הספר (לפני הקטלוג המצורף) ומתקדם פנימה אל תוך הספר.

לאחר הנספחים תוכל למצוא את האינדקס. הפניות האינדקס הן שתיים: אחת לפי מספרי העמודים של הפרקים בעברית - הם יסומנו במספרים, והשנייה לפי המיקום - Appendix A (נספח א') או Appendix B (נספח ב').

Appendixes

Appendix A

ASP Objects and Components Reference

In This Appendix

- **ASP Built-In objects**
- **ASP Installable Components**

This appendix provides a quick reference for the Active Server Page objects and installable components. This reference is not intended to be comprehensive. For more information on each object and component, consult the Microsoft online documentation (available at http://localserver/iishelp on a server with IIS installed).

ASP Built-In Objects

You can use any of the following Active Server Pages built-in objects without explicitly declaring them in your Active Server Page scripts.

The Application Object

The Application object can be used to store and retrieve information that can be shared by all users of an application.

Collections

Contents(key)	Contains data and objects with application scope not declared with the <object> tag
StaticObjects(key)	Contains objects with application scope declared with the <object> tag

Methods

Lock	Prevents other users from accessing the Application collection
Unlock	Enables other users to access the Application collection

Events

OnEnd	Triggered by stopping the Web server, modifying the Global.asa file, or unloading the application
OnStart	Triggered by the first request for a Web page in the application

The ObjectContext Object

The ObjectContext object is used to control Active Server Page transactions.

Methods

SetAbort	Explicitly aborts a transaction
SetComplete	Overrides any calls to the SetAbort method

Events

OnTransactionAbort	Triggered by Aborted transaction
OnTransactionCommit	Triggered by Completed transaction

The Request Object

The Request object represents all information sent in a request from a browser to your server.

Collections

ClientCertificate(key)	Contains values from client-certificate
Cookies(key)	Contains values from browser cookies
Form	Contains values from HTML form fields
QueryString	Contains values from query string
ServerVariables	Contains values from headers and environmental variables

Method

BinaryRead(count)	Retrieves raw content of HTML form

Property

TotalBytes	The number of bytes in the browser request

The Response Object

The Response object represents all information that is sent from your server to a browser.

Collection

Cookies(key)	Enables you to add a cookie to a browser

Methods

AddHeader name, value	Adds new header to a browser
AppendToLog	Adds an entry to the IIS log file

Methods

BinaryWrite data	Writes non-string information in the response
Clear	Clears the buffered response
End	Ends processing of a script
Flush	Sends all contents of the buffer
Redirect URL	Redirects a browser to a new page
Write variant	Sends a string to a browser.

Properties

Buffer	Buffers an Active Server Page
CacheControl	Controls caching by proxy servers
CharSet(Charsetname)	Character set to use on current page
ContentType	Specifies the content type of the response
Expires	Controls caching by browsers using relative time
ExpiresAbsolute	Controls caching by browsers using absolute time
IsClientConnected	Indicates whether a browser is still connected
PICS(PICSLABEL)	Adds a PICS rating to a page
Status	Specifies the status line returned by server

The Server Object

The Server object enables the use of various utility functions on the server.

Methods

CreateObject(progID)	Creates an instance of an object
HTMLEncode(string)	Converts a string to use special HTML characters
MapPath(path)	Converts a virtual path to a physical path
URLEncode(string)	Converts a string to URL-encoded form

Property

ScriptTimeout	The number of seconds a script is allowed to run before being terminated

The Session Object

The Session object represents the information about a particular user session.

Collections

Contents(key)	Contains data and objects with session scope declared without using the <object> tag
StaticObjects(key)	Contains objects with session scope declared using the <object> tag

Method

Abandon	Ends a user session after the current page is processed

Properties

CodePage	Specifies the server code page
LCID	Specifies the location identifier
SessionID	A unique identifier for the user session
Timeout	The time in minutes before a session terminates

Events

OnEnd	Triggered by the user not requesting a page longer than the Session Timeout period
OnStart	Triggered by the first request for a Web page by the user

ASP Installable Components

Before using any of the following installable components in your Active Server Page scripts, you must explicitly declare an instance of the component. A sample declaration is provided for each component.

The Ad Rotator Component

The Ad Rotator component enables you to display banner advertisements with different frequencies. To create an instance of the Ad Rotator component use

```
<% Set MyAd=Server.CreateObject("MSWC.AdRotator") %>
```

Method

GetAdvertisement	Displays an advertisement on a page
(Rotator Schedule File)	

Properties

Border(size)	Displays a border around the advertisement with a size in pixels
Clickable(boolean)	Specifies whether an advertisement is a hyperlink
TargetFrame(frame)	The HTML frame where the advertisement should be displayed

Files

Redirection File	When you click an advertisement, you're sent to this page
Rotator Schedule File	Text file that lists the properties of the advertisements

The Browser Capabilities Component

The Browser Capabilities component enables you to detect the features of a Web browser. To create an instance of the Browser Capabilities component use

```
<% Set MyBrow=Server.CreateObject("MSWC.BrowserType") %>
```

File

Browscap.ini	Text file that contains information about browsers

The Content Linking Component

To create an instance of the Content Linking component use

```
<% Set myLink=Server.CreateObject("MSWC.NextLink") %>
```

Methods

GetListCount(Content Linking List File)	Returns total number of pages in the Content Linking List File
GetListIndex(Content Linking List File)	Returns the position of the current page in the Content Linking List File
GetNextDescription (Content Linking List File)	Returns the description of the next page in the Content Linking List File
GetNextURL(Content Linking List File)	Returns the URL of the next page in the Content Linking List File
GetNthDescription (Content Linking List File, index)	Returns the description of the page with the index in the Content Linking List File

Methods

GetNthURL (Content Linking List File, index)	Returns the URL of the page with the index in the Content Linking List File
GetPreviousDescription (Content Linking List File)	Returns the description of the previous page in the Content Linking List File
GetPreviousURL (Content Linking List File)	Returns the URL of the previous page in the Content Linking List File

File

Content Linking List File	Text file containing a list of pages

The Content Rotator Component

The Content Rotator component enables you to display HTML strings with different frequencies. To create an instance of the Content Rotator component use

```
<% Set myContent=Server.CreateObject("IISSample.ContentRotator") %>
```

Methods

ChooseContent(Content Schedule File)	Returns an HTML string from the Content Schedule File
GetAllContent(Content Schedule File)	Returns all HTML strings from the Content Schedule File

File

Content Schedule File	Text file that contains a list of HTML strings

The Counters Component

The Counters component enables you to create one or more counters to track information such as the number of visits to a Web site. To create an instance of the Counters Component include the following HTML code in the Global.asa file:

```
<OBJECT RUNAT=SERVER SCOPE=Application ID=myCounter
PROGID="MSWC.Counters"></OBJECT>
```

Methods

Get(counter)	Returns the value of a counter
Increment(counter)	Adds one to the value of a counter
Remove(counter)	Removes a counter
Set counter, number	Sets counter to equal a certain numeric value

File	
Counters.txt	This text file contains all the individual counters

The Lookup Table Component

The Lookup Table component enables you to cache information in an object that can be created with page, session, or application scope. To create an instance of the Lookup Table component with page scope use the following:

```
<% SET myLookup = Server.CreateObject( "IISSample.LookupTable" )%>
```

To create an instance of the Lookup Table component with application scope add the following code to the Global.asa file:

```
<OBJECT RUNAT=SERVER SCOPE=Application ID=myLookup
PROGID="IISSample.LookupTable"></OBJECT>
```

Methods	
Count	Returns the number of key/value pairs that exist in the Lookup table.
LoadValuesfilepath, format	Loads key/value pairs into the component from a file. See the values for format.
LoadValuesExfilepath, format, period	Loads key/value pairs into the component from a file on a periodic basis. See the values for format. The value for period is specified in seconds.

Methods	
LookupValue(key)	Returns the value for a specific key.
Key(index)	Returns the key located at the index position.
KeyExists(key)	Returns True if the specified key exists in the Lookup Table.
ReadLock	Locks the Lookup Table until the ReadUnlock method is called.
ReadUnlock	Releases a lock created by calling ReadLock.
Value(index)	Returns the value of the key located at the index position.

File

Lookup Table File	Text file that stores the key/value pairs. Each key/value pair is separated by a comma. To create multiline values, use \ at the end of a line as a continuation character.

Format

0	String keys and string values.
1	String keys and integer values.
2	Integer keys and string values.
3	Integer keys and integer values.
10	String keys and string values (ignore duplicate keys).
11	String keys and integer values (ignore duplicate keys).
12	Integer keys and string values (ignore duplicate keys).
13	Integer keys and integer values (ignore duplicate keys).

The NewMail Object

The NewMail object enables you to send email from an Active Server Page. To create an instance of the NewMail object use

```
<%SET myMail = Server.CreateObject("CDONTS.Newmail")%>
```

Methods

AttachFile(file path)	Attaches a file to an email message
AttachURL(file path, URL)	Attaches a file to an email message with an associated URL
Send	Sends an email message

Properties

Bcc	The email address used for sending a blind carbon copy of the email message
Body	The text of the email message
BodyFormat	Specifies whether the email message contains HTML or plain text

Properties

Cc	The email address used for sending a carbon copy of the email message
ContentBase	Sets the base URL for all URLs included in the email message
ContentLocation	Sets an absolute or relative path for all URLs included in the email message
From	The email address of the sender of the email message
Importance	The priority of the email message
MailFormat	The encoding format of the email message
Subject	The subject line of the email message
To	The email address of the recipient of the email message

The Page Counter Component

The Page Counter component enables you to track the number of times that a page has been requested. To create an instance of the Page Counter component use

```
<% Set myCNT=Server.CreateObject("IISSample.PageCounter")%>
```

Methods

Hits([Web Page Path])	Returns the number of times a page has been requested
PageHit()	Updates the number of times the current page has been requested
Reset([Web Page Path])	Resets to zero the number of times a page has been requested

File

Hit Count Data File	Text file that stores the number of requests that each page has received

Appendix B

VBScript Reference

In This Appendix

- **Statements**
- **Functions**
- **Operators**
- **Objects**
- **Constants**

This appendix contains a quick reference of all the VBScript statements, functions, operators, and objects. Many of the VBScript functions use constants. See the end of this appendix for a list of these constants.

For more information on VBScript, consult the Microsoft VBScript online documentation included with the Windows NT Option Pack or visit the Microsoft Windows Script Technologies Web site at http://msdn.microsoft.com/scripting.

Statements

Call

[Call] *name* [*argumentlist*]

This statement transfers control to a function or subroutine. Using Call when calling a function or subroutine is always optional. However, if the optional Call keyword is used, *argumentlist* must be enclosed in parentheses.

Class

Class *name*
[*statements*]
End Class

Defines a class, assigning a name, variables, properties, and methods.

Const

[Public | Private] Const *constantname=expression*

This statement is used to declare a constant. You can declare multiple constants within a single line by separating each constant assignment with a comma.

Dim

Dim *varname*[([*subscripts*])][, *varname*[([*subscripts*])]] . . .

This statement creates a new variable and allocates storage space.

Do...Loop

Syntax 1:
Do [{While | Until} *condition*]
[*statements*]
[Exit Do]
[*statements*]
Loop
Syntax 2:

Do
[*statements*]
[Exit Do]
[*statements*]
Loop [{While | Until} *condition*]

Both forms of the statement repeat *statements* while *condition* is TRUE or until it becomes TRUE.

Erase

Erase *array*

This statement erases *array*, reinitializing elements of fixed-size arrays and recovering storage space of dynamic arrays.

Execute

Execute *statement*

Executes a statement at runtime. The statement can contain multiple statements divided by colons.

Exit

Exit Do
This statement exits a Do...Loop statement.

Exit For
This statement exits a For...Next or For Each...Next loop.

Exit Function
This statement exits a function.

Exit Sub
This statement exits a subroutine.

For...Next

For *counter* = *start* To *end* [Step *step*]
[*statements*]
[Exit For]
[*statements*]
Next

This loop repeats a group of *statements* the number of times designated by the loop counter.

For Each...Next

For Each *element* In *group*
[*statements*]
[Exit For]
[*statements*]
Next [*element*]

For each element in the array or collection, this loop repeats the group of *statements*.

Function

[Public | Private] Function *name* [(*arglist*)]
[*statements*]
[name = expression]
[Exit Function]
[*statements*]
[*name = expression*]
End Function

This statement defines a Function, assigning a name, arguments, and code.

If...Then...Else

Syntax 1:

If *condition* Then *statements* [Else *elsestatements*]

Syntax 2:

If *condition* Then
statements
[ElseIf *condition-n* Then
[*elseifstatements*]] . . .
[Else
[*elsestatements*]]
End If

Both forms of the statement conditionally execute groups of statements.

On Error

On Error Resume Next

When an error occurs, this statement executes the statement immediately following the statement that caused the runtime error, or executes the statement immediately following the most recent call out of the procedure containing the On Error Resume Next statement. To re-enable the reporting of errors, use On Error Goto 0.

Option Explicit

Option Explicit

This statement forces the explicit declaration of all variables using the Dim, Private, Public, or ReDim statements. It must appear before any other statement or content in the Active Server Page.

Private

Private *varname*[([*subscripts*])][, *varname*[([*subscripts*])]] . . .

This statement creates private variables (variables available only to the script in which they were declared) and allocates storage space.

Property Get

[Public [Default]| Private] Property Get *name* [(*arglist*)]
[*statements*]
[[Set] *name* = *expression*]
[Exit Property]
[*statements*]
[[Set] *name* = *expression*]
End Property

Defines a property procedure that returns the value of a property for a class.

Property Let

[Public | Private] Property Let *name* ([*arglist,*] value)
[*statements*]
[Exit Property]
[*statements*]
End Property

Defines a property procedure that sets the value of a property for a class.

Property Set

[Public | Private] Property Set *name*([*arglist,*] *reference*)
[*statements*]
[Exit Property]
[*statements*]
End Property

Defines a property procedure that sets a reference to an object for a class.

Public

Public *varname*[([*subscripts*])][, *varname*[([*subscripts*])]] . . .

This statement creates public variables (variables available to all procedures in all scripts in all projects) and allocates storage space.

Randomize

Randomize [*number*]

This statement gives the Rnd function's random-number generator a new seed value. If you use this statement without supplying a number, the system timer is used for the seed value. If you do not use Randomize before using the RND() function, the same sequence of numbers will be generated every time RND() is called.

ReDim

ReDim [Preserve] *varname*(*subscripts*) [, *varname*(*subscripts*)] . . .

This statement revises dimension subscripts, sizing or resizing a dynamic array. Preserve protects data in the existing array.

Rem

Syntax 1:
Rem *comment*

Syntax 2:
' comment

Both forms of this statement keep comments from being parsed. If Rem follows other statements on a line, it must be separated by a colon.

Select Case

Select Case *testexpression*
[Case *expressionlist-n*
[*statements-n*]] . . .
[Case Else *expressionlist-n*
[*elsestatements-n*]]
End Select

This statement executes statements paired with any *expressionlist* that matches testexpression. If *testexpression* doesn't match any *expressionlist*, the statements paired with Case Else are executed.

Set

Set *objectvar* = {*objectexpression* | Nothing}

This statement sets the object reference for the variable or the property. Nothing dissociates *objectvar* from any specific object.

Sub

[Public | Private] Sub *name* [(*arglist*)]
[*statements*]
[Exit Sub]
[*statements*]
End Sub

This statement defines a subroutine, assigning a name, arguments, and code.

While...Wend

While *condition*
[*statements*]
Wend

This loop continues to execute the series of *statements* as long as *condition* is TRUE.

With

With *object*
[*statements*]
End With

With enables you to perform a series of statements with a single object without explicitly referring to the object in every statement.

Functions

Abs(*number*)

This function returns an absolute value.

Array(*arglist*)

This function creates an array.

Asc(*string*)

This function returns the ANSI code for the first letter in *string*.

AscB(*string*)

This function returns the first byte in *string*.

AscW(*string*)

This function returns the first Unicode character code in *string*.

Atn(*number*)

This function returns the arctangent.

CBool(*expression*)

This function converts to variant of subtype Boolean.

CByte(*expression*)

This function converts to variant of subtype Byte.

CCur(*expression*)

This function converts to variant of subtype Currency.

CDate(*date*)

This function converts to variant of subtype Date.

CDbl(*expression*)

This function converts to variant of subtype Double.

Chr(*charcode*)

This function converts the ANSI character code to the corresponding keyboard character.

ChrB(*charcode*)

This function converts the character code to a single byte.

ChrW(*charcode*)

This function converts the Unicode character code to the corresponding keyboard character.

CInt(*expression*)

This function converts to variant of subtype Integer.

CLng(*expression*)

This function converts to variant of subtype Long.

Cos(*number*)

This function returns the cosine.

CreateObject(*servername.typename*)

This function creates an Automation object. Within an Active Server Page, use the CreateObject method of the Server object instead of this function.

CSng(*expression*)

This function converts to variant of subtype Single.

CStr(*expression*)

This function converts to variant of subtype String.

Date

This function returns the date according to the system.

DateAdd(*interval,number,date*)

This function adds the time interval to date. The interval argument accepts the following values:

Setting	Description
yyyy	Year
q	Quarter
m	Month
y	Day of Year
d	Day
w	Weekday
ww	Week of Year
h	Hour
n	Minute
s	Second

DateDiff(*interval,date1,date2*[, *firstdayofweek*[,*firstweekofyear*]])

This function returns the number of intervals between two dates. See the later section "Date and Time Constants" for values of *firstdayofweek* and *firstweekofyear*. The *interval* argument accepts the values shown in the following table:

Setting	Description
yyyy	Year
q	Quarter
m	Month
y	Day of year
d	Day
w	Weekday
ww	Week of year
h	Hour
n	Minute
s	Second

DatePart(*interval,date*[,*firstdayofweek*[, *firstweekofyear*]])

This function returns the designated part of the date. See the later section "Date and Time Constants" for values of *firstdayofweek* and *firstweekofyear*. The *interval* argument accepts the values shown in the following table:

Setting	Description
yyyy	Year
q	Quarter
m	Month
y	Day of year
d	Day
w	Weekday
ww	Week of year
h	Hour
n	Minute
s	Second

DateSerial(*year,month,day*)

This function converts to variant of subtype Date.

DateValue(*date*)

This function converts to variant of subtype Date.

Day(*date*)

This function returns a number representing the day of the month according to the argument.

Exp(*number*)

This function raises e to the power of *number*.

Eval(*expression*)

This function evaluates the expression and returns the result.

Filter(*InputStrings,Value*[,*Include*[, *Compare*]])

This function creates a new array according to the filter criteria. See the later section "Comparison Constants" for values of *Compare*. *InputStrings* must be a one-

dimensional array of strings. Value is the string to search for.

Fix(*number*)

This function converts to integer (for a negative number, this function rounds the number up).

FormatCurrency(*Expression*[,*NumDigitsAfterDecimal*[,*IncludeLeadingDigit*[,*UseParensForNegativeNumbers*[,*GroupDigits*]]]])

This function formats an expression as currency. See the later section "Tristate Constants" for values of *IncludeLeadingDigit*, *UseParensForNegativeNumbers*, and *GroupDigits*.

FormatDateTime(*Date*[,*NamedFormat*])

This function formats dates and times. See the later section "Date Format Constants" for values of *NamedFormat*.

FormatNumber(*Expression*[,*NumDigitsAfter Decimal*[,*IncludeLeadingDigit*[,*UseParensForNegativeNumbers*[,*GroupDigits*]]]])

This function formats numbers. See the later section "Tristate Constants" for values of *IncludeLeadingDigit*, *UseParensForNegativeNumbers*, and *GroupDigits*.

FormatPercent(*Expression*[,*NumDigitsAfter Decimal*[,*IncludeLeadingDigit*[,*UseParensForNegativeNumbers*[,*GroupDigits*]]])

This function formats percentages. See the later section "Tristate Constants" for values of *IncludeLeadingDigit*, *UseParensForNegativeNumbers*, and *GroupDigits*.

GetLocale

This function returns the current locale (as a 32-bit integer) without setting a new one. The locale determines the formatting of currencies and dates.

GetObject([*pathname*][,*class*])

This function returns the specified Automation object from the specified file.

Hex(*number*)

This function returns the hexadecimal value of *number*.

Hour(*time*)

This function returns the hour according to the *time* argument.

InputBox(*prompt*[,*title*][,*default*][,*xpos*][,*ypos*][,*helpfile*,*context*])

This function prompts and returns user input. Don’t use this with Active Server Pages!

InStr([*start,*]*string1,string2*[,*compare*])

This function returns the first appearance of *string2* within *string1*. See the later section "Comparison Constants" for values of *compare*.

InStrB([*start,*]*string1,string2*[,*compare*])

This function returns the byte position of the first appearance of *string2* within *string1*. See the later section "Comparison Constants" for values of *compare*.

InStrRev(*string1,string2*[*start*[,*compare*]])

This function returns the first appearance of *string2* within *string1* starting from the end of the string. See the later section "Comparison Constants" for values of *compare*.

Int(*number*)

This function returns an integer (for a negative number, it rounds the number down).

IsArray(*varname*)

This function determines whether the variable is an array.

IsDate(*expression*)

This function determines whether *expression* can be converted to date format.

IsEmpty(*expression*)

This function determines whether the variable has been initialized.

IsNull(*expression*)

This function determines whether *expression* is null.

IsNumeric(*expression*)

This function determines whether *expression* is a number.

IsObject(*expression*)

This function determines whether *expression* is an Automation object.

Join(*list*[,*delimiter*])

This function joins substrings in an array separated by the character indicated by *delimiter*. It's the opposite of the Split function.

LBound(*arrayname*[,*dimension*])

This function returns the lower limit of the array dimension. It always returns zero with the current version of VBScript.

LCase(*string*)

This function converts a string to lowercase.

Left(*string*,*length*)

This function returns the left string portion of the designated length.

LeftB(*string*, *bytes*)

This function returns the left string portion of the designated number of bytes.

Len(*string* | *varname*)

This function returns the length of the string or the byte size of the variable.

LenB(*string*)

This function returns the number of bytes used to represent the string.

LoadPicture(*picturename*)

This function loads a picture object.

Log(*number*)

This function returns the natural logarithm of the number.

LTrim(*string*)

This function removes extra leading spaces.

Mid(*string*,*start*[,*length*])

This function returns a string portion of the designated length.

MidB(*string*,*start*[,*bytes*])

This function returns a string portion of the designated number of bytes.

Minute(*time*)

This function returns the minute according to the *time* argument.

Month(*date*)

This function returns the month represented by the number.

MonthName(*month*[,*abbreviate*])

This function returns the month represented by the name.

MsgBox(*prompt*[,*buttons*][,*title*][*helpfile*, *context*])

This function prompts the user to choose a button and indicates which button the user has chosen. See the later section "MsgBox Constants" for *button* and return values. Don't use this in an Active Server Page!

Now

This function returns the current date and time according to the system.

Oct(*number*)

This function returns the octal value of *number*.

Replace(*expression*,*find*,*replacewith*[,*start*[,*count*[, *compare*]]])

This function replaces the designated substring *find* with the substring *replacewith* the designated number of times. See the later section "Comparison Constants" for values of *compare*.

Right(*string*,*length*)

This function returns the right string portion of the designated length.

RightB(*string,bytes*)

This function returns the right string portion of the designated number of bytes.

Rnd([*number*)])

This function generates a pseudo-random number.

Round(*number*[,*numdecimalplaces*])

This function rounds *number*.

RTrim(*string*)

This function removes extra right spaces.

ScriptEngine

This function returns the name of the scripting language in use.

ScriptEngineBuildVersion

This function returns the name of the script engine in use.

ScriptEngineMajorVersion

This function returns the major version number of the script engine in use.

Second(*time*)

This function returns the second of the minute according to the *time* argument.

SetLocale(*lcid*)

This function returns the current locale and sets the new locale specified by *lcid*. After the locale is set, currencies and dates are formatted according to the locale.

Sgn(*number*)

This function returns the sign of *number*.

Sin(*number*)

This function returns the sine of *number*.

Space(*number*)

This function creates a string with the specified number of spaces.

Split(*expression*[,*delimiter*[,*count*[, *compare*]]])

This function splits a string and converts it into an array. It's the opposite of the Join function. See the later section "Comparison Constants" for values of *compare*.

Sqr(*number*)

This function returns the square root of the specified number.

StrComp(*string1*,*string2*[,*compare*])

This function performs a string comparison. See the later section "Comparison Constants" for values of *compare*.

StrReverse(*string1*)

This function reverses the characters of a string.

String(*number*,*character*)

This function creates a string with character repeated the specified number of times.

Tan(*number*)

This function returns the tangent of *number*.

Time

This function returns the current time according to the system.

TimeSerial(*hour*,*minute*,*second*)

This function returns Date Variant.

TimeValue(*time*)

This function returns Date Variant containing time.

Trim(*string*)

This function removes extra spaces at left and right.

TypeName(*varname*)

This function returns the subtype by name. See the later section "VarType Constants" for return values.

UBound(*arrayname*[,*dimension*])

This function returns the upper bound of the array dimension. If no dimension is specified, the first dimension is assumed.

UCase(*string*)

This function converts the string to uppercase.

VarType(*varname*)

This function returns the subtype by value. See the later section "VarType Constants" for return values.

Weekday(*date*,[*firstdayofweek*])

This function returns the day of the week by number. See the later section "Date and Time Constants" for values of *firstdayofweek*.

WeekDayName(*weekday*,*abbreviate*, *firstdayofweek*)

This function returns the day of the week by name. See the later section "Date and Time Constants" for values of *firstdayofweek*.

Year(*date*)

This function returns the year according to the argument.

Operators

+ Operator

result = *expression1* + *expression2*

Sum. This operator can also be used for string concatenation but & is less ambiguous.

And Operator

result = *expression1* And *expression2*

Logical conjunction. This operator also performs bitwise comparison, returning digit 1 only where both expressions have digit 1.

& Operator

result = *expression1* & *expression2*

String concatenation.

/ Operator

result = *number1*/*number2*

This operator divides two numbers and returns a floating-point number.

Eqv Operator

result = *expression1* Eqv *expression2*

Logical equivalence. This operator also performs bitwise comparison, returning digit 1 only where bits in the two expressions are identical.

^ Operator

result = *number*^*exponent*

This operator raises a number to the power of exponent.

Imp Operator

result = *expression1* Imp *expression2*

Material implication. This operator also performs bitwise comparison. The following table illustrates logical implication:

expression1	expression2	result
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
TRUE	NULL	NULL
FALSE	TRUE	TRUE
FALSE	FALSE	TRUE
FALSE	NULL	TRUE
NULL	TRUE	TRUE
NULL	FALSE	NULL
NULL	NULL	NULL

The following table illustrates the bitwise comparison associated with the Imp operator:

expression1	expression2	result
0	0	1
0	1	1
1	0	0
1	1	1

\ Operator

result = number1\number2

This operator divides two numbers and returns an integer.

Is Operator

result = object1 Is *object2*

This operator checks whether two variables refer to the same object and returns TRUE or FALSE.

Mod Operator

result = number1 Mod *number2*

This operator divides two numbers and returns only the remainder.

* Operator

result = number1 number2*

Multiplication.

- Operator

result = *number1-number2*

Subtraction.

-number

Sign of number.

Not Operator

result = Not *expression*

Logical negation. This operator also performs bitwise negation.

Or Operator

result = *expression1* Or *expression2*

Logical disjunction. *result* is TRUE if either expression is TRUE or both expressions are TRUE. The Or operator also performs bitwise comparison, returning digit 0 only where both expressions have the digit 0. Elsewhere the digit 1 is returned.

Xor Operator

result = *expression1* Xor *expression2*

Logical exclusion. *result* is TRUE if one and only one expression is TRUE. The Xor operator also performs bitwise comparison, returning the digit 1 only where one and only one expression has the digit 1. Elsewhere the digit 0 is returned.

Objects

The Dictionary Object

The Dictionary object can be used to store key/value pairs. To use the Dictionary object in an Active Server Page script, you must declare an instance of the object like this:

<% Set myDict = Server.CreateObject("Scripting.Dictionary")%>

Methods

Add *Key, Item*

This method adds *Key* and associated *Item* to a Dictionary object.

Exists(*Key*)
This method checks whether the specified *Key* already exists in the Dictionary object and returns either TRUE or FALSE.

Items
This method returns all *Items* in the Dictionary object as an array.

Keys
This method returns all existing *Keys* in the Dictionary object as an array.

Remove(*Key*)
This method removes *Key* and its associated item from the Dictionary object.

RemoveAll
This method removes all keys and their associated items from the Dictionary object.

Properties

CompareMode
This property specifies how items in the Dictionary should be compared. See the later section "Comparison Constants" for possible values.

Count
This property counts items in the Dictionary object (read-only).

Item(*Key*)
This property returns the item associated with the designated *Key* in the Dictionary object or associates a new value with the key.

Key(*Key*)
This property sets the specified *Key* in the Dictionary object.

The Err Object
The Err object contains information about runtime errors and enables you to raise custom errors. You do not need to create an instance of this object to use it in your scripts.

Methods

Clear
This method explicitly clears the Err object of all property settings.

Raise(*number*, *source*, *description*, *helpfile*, *helpcontext*)

This method returns a runtime error. number identifies the type of error. VBScript errors are numbered in the range 0–65535. *source* indicates the object or application that originally generated the error.

Properties

Description

This property returns or sets a brief description of an error.

HelpContext

This property sets or returns the identifier for a topic within the Help file that's appropriate for the Err object.

HelpFile

This property sets or returns the fully qualified path to the Help file that's appropriate for the Err object.

Number

This property returns a number identifying the error or, if *errornumber* is included, associates the error with either a VBScript error number or an SCODE error value.

Source

This property identifies the source of the error, usually by class name or programmatic ID of the object or the application that generated the error.

The Match Object

The Match object represents the properties of a match that results from calling the Execute method of the RegExp object. You never explicitly create an instance of the Match object.

Properties

FirstIndex

This property returns the position where the first match occurs in the string passed to the Execute method of the RegExp object.

Length

This property returns the length of the match.

Value

This property returns the actual value of the match.

The Matches Collection

The Matches collection contains all the Match objects returned by calling the Execute method of the RegExp object.

Properties

Count

This property returns the number of Match objects in the Matches collection.

Item(*key*)

This property returns the Match object specified by *key*.

The RegExp Object

The RegExp object enables you to perform regular expression pattern matching. To use the RegExp object in an Active Server Page, you must declare an instance of it like this:

```
<% Set myReg = Server.CreateObject( "VBScript.RegExp" ) %>
```

Methods

Execute(*string*)

This method performs pattern matching against the *string* using the pattern set with the Pattern property.

Replace(*string1*, *string2*)

This method replaces the text in *string1* with the text in *string2* according to the pattern set with the Pattern property.

Test(*string*)

This method returns TRUE if *string* matches the pattern set with the Pattern property.

Properties

Global

If this property is assigned the value TRUE, every match is returned. Otherwise, only the first match is returned.

IgnoreCase

If this property is assigned the value TRUE, then a case-insensitive pattern match is performed.

Pattern

This property is used to set or return the regular expression pattern that is used when performing matches. The pattern can contain any of the regular expression characters in the following table:

Character	Description
\	Matches special characters such as the newline character (\n) or escapes regular expression character (\\)
^	Matches the beginning of input
$	Matches the end of input
*	Matches the character that immediately precedes it if it appears zero or more times
+	Matches the character that immediately precedes it if it appears one or more times
?	Matches the character that immediately precedes it if it appears zero times or one time
.	Matches any single character except the newline character
(pattern)	Matches the pattern in the parentheses and remembers the match in the Matches collection
x|y	Matches either character *x* or character *y*
{n}	Matches the character that immediately precedes it if it appears exactly *n* times
{n,m}	Matches the character that immediately precedes it if it appears at least *n* times and at most *m* times
[xyz]	Matches any one of the characters in the brackets
[^xyz]	Matches any character not in the brackets
[a-z]	Matches any character in the range of characters
[^a-z]	Matches any character not in the range of characters
\b	Matches a word boundary
\B	Matches a nonword boundary
\d	Matches a numeral
\D	Matches any non-numeric character
\f	Matches form feed character
\n	Matches newline character

Character	Description
\r	Matches carriage return character
\s	Matches whitespace
\S	Matches nonwhitespace character
\t	Matches a tab character
\v	Matches a vertical tab character
\w	Matches any word character or underscore
\W	Matches any nonword character
\(num)	Matches previously remembered pattern if it appears *num* times
\n	Matches ASCII character (*n* represents an ASCII character as an octal escape value)
\xn	Matches ASCII character (*n* represents an ASCII character as a hexadecimal escape value)

Constants

In the following sections, the constants (left column of each table) can be used in place of the values (in the center column).

Color Constants

Constant	Value	Description
vbBlack	&h00	Black
vbRed	&hFF	Red
vbGreen	&hFF00	Green
vbYellow	&hFFFF	Yellow
vbBlue	&hFF0000	Blue
vbMagenta	&hFF00FF	Magenta
vbCyan	&hFFFF00	Cyan
vbWhite	&hFFFFFF	White

Comparison Constants

Constant	Value	Description
vbBinaryCompare	0	Binary comparison
vbTextCompare	1	Textual comparison
vbDatabaseCompare	2	Comparison based on information in the database

Date and Time Constants

Constant	Value	Description
vbSunday	1	Sunday
vbMonday	2	Monday
vbTuesday	3	Tuesday
vbWednesday	4	Wednesday
vbThursday	5	Thursday
vbFriday	6	Friday
vbSaturday	7	Saturday
vbFirstJan1	1	Week of Jan 1 (default)
vbFirstFourDays	2	First week of the year that has at least four days
vbFirstFullWeek	3	First full week of the year
vbUseSystem	0	Use the date format of the computer's regional settings
vbUseSystemDayOfWeek	0	Use the first day of the week according to the system settings

Date Format Constants

Constant	Value	Description
vbGeneralDate	0	Display the date and/or time according to the system settings
vbLongDate	1	Display the date in long date format
vbShortDate	2	Display the date in short date format
vbLongTime	3	Display the time in long time format
vbShortTime	4	Display the time in short time format

MsgBox Constants

These constants are used with the MsgBox function to specify buttons and icons displayed in the message box and to identify the default icon.

Constant	Value	Description
vbOKOnly	0	Show only the OK button.
vbOKCancel	1	Show OK and Cancel buttons.
vbAbortRetryIgnore	2	Show Abort, Retry, and Ignore buttons.
vbYesNoCancel	3	Show Yes, No, and Cancel buttons.

Constant	Value	Description
vbYesNo	4	Show Yes and No buttons.
vbRetryCancel	5	Show Retry and Cancel buttons.
vbCritical	16	Show Critical Message icon.
vbQuestion	32	Show Warning Query icon.
vbExclamation	48	Show Warning Message icon.
vbInformation	64	Show Information Message icon.
vbDefaultButton1	0	The first button is the default.
vbDefaultButton2	256	The second button is the default.
vbDefaultButton3	512	The third button is the default.
vbDefaultButton4	768	The fourth button is the default.

The constants in the following table specify modality:

Constant	Value	Description
vbApplicationModal	0	The current application won't continue until the user responds to the message box.
vbSystemModal	4096	No applications will continue until the user responds to the message box.

The constants in the following table identify which button has been pressed:

Constant	Value	Description
vbOK	1	OK button
vbCancel	2	Cancel button
vbAbort	3	Abort button
vbRetry	4	Retry button
vbIgnore	5	Ignore button
vbYes	6	Yes button
vbNo	7	No button

String Constants

Constant	Value	Description
vbCr	Chr(13)	Carriage return
vbCrLf	Chr(13) & Chr(10)	Combination carriage return and line feed
vbFormFeed	Chr(12)	Form feed
vbLF	Chr(10)	Line feed

Constant	Value	Description
vbNewLine	Chr(13)&Chr(10) or Chr (10)	Newline character appropriate for platform
vbNullChar	Chr(0)	Character of value 0
vbNullString	string having value 0	Null string
vbTab	Chr(9)	Horizontal tab
vbVerticalTab	Chr(11)	Vertical tab

Tristate Constants

Constant	Value	Description
TristateTrue	-1	True
TristateFalse	0	False
TristateUseDefault	-2	Use the default setting

VarType Constants

Constant	Value	Description
vbEmpty	0	Uninitialized (default)
vbNull	1	Contains no valid data
vbInteger	2	Integer subtype
vbLong	3	Long subtype
vbSingle	4	Single subtype
vbDouble	5	Double subtype
vbCurrency	6	Currency subtype
vbDate	7	Date subtype
vbString	8	String subtype
vbObject	9	Object
vbError	10	Error subtype
vbBoolean	11	Boolean subtype
vbVariant	12	Variant (only used for arrays of variants)

Constant	Value	Description
vbDataObject	13	Data access object
vbDecimal	14	Decimal subtype
vbByte	17	Byte subtype
vbArray	8192	Array

Index

O

P

R

S

W